

**VŠB – TECHNICKÁ UNIVERZITA OSTRAVA**

**FAKULTA STROJNÍ**

**KATEDRA ROBOTOTECHNIKY**

**VÝUKOVÉ PRACOVISTĚ S PRŮMYSLOVÝM ROBOTEM IRB140**

**EDUCATIONAL WORKPLACE WITH IRB140 INDUSTRIAL ROBOT**

**Student:**

**Bc. Ondřej Seidler**

**Vedoucí diplomové práce:**

**prof. Dr. Ing. Petr Novák**

**OSTRAVA 2013**

## Zadání diplomové práce

Student: **Bc. Ondřej Seidler**  
Studijní program: N2301 Strojní inženýrství  
Studijní obor: 2301T013 Robotika  
Téma: **Výukové pracoviště s průmyslovým robotem IRB140**  
**Educational Workplace with IRB140 Industrial Robot**

Zásady pro vypracování:

1. Popište dané pracoviště s PR ABB IRB140.
2. Navrhněte a realizujte demonstrační a výukové úlohy s na pracovištích s průmyslovými roboty ABB IRB 140.
3. Práci doplňte podrobnou dokumentací k realizovaným úlohám.
4. Práci též doložte v elektronické podobě ve formátu MS WORD

Seznam doporučené odborné literatury:

ČSN 01 6910 *Úprava písemností psaných strojem nebo zpracovaných textovými editory*. Praha: Český normalizační institut, srpen 2007. 48 s.

ČSN ISO 690 *Informace a dokumentace - Pravidla pro bibliografické odkazy a citace informačních zdrojů*. Praha: Český normalizační institut, 2010.

Manuály ABB dodané s robotizovaným pracovištěm.


Chamrad, T. *Aplikace strojového vidění pro robotizovaná technologická pracoviště*. diplomová práce, VŠB -TU Ostrava, Katedra robototechniky, 2013, 122s.

Formální náležitosti a rozsah diplomové práce stanoví pokyny pro vypracování zveřejněné na webových stránkách fakulty.

Vedoucí diplomové práce: **prof. Dr.Ing. Petr Novák**

Datum zadání: 31.10.2013

Datum odevzdání: 19.05.2014



prof. Dr.Ing. Petr Novák  
vedoucí katedry



doc. Ing. Ivo Hlavatý, Ph.D.  
děkan fakulty

Místopřísežné prohlášení studenta

Prohlašuji, že jsem celou diplomovou práci včetně příloh vypracoval samostatně pod vedením vedoucího diplomové práce a uvedl jsem všechny použité podklady a literaturu.

16.5.2014

V Ostravě .....



.....

Bc. Ondřej Seidler

Prohlašuji, že

- jsem celou diplomovou práci včetně příloh vypracoval samostatně pod vedením vedoucího diplomové práce a uvedl jsem všechny použité podklady a literaturu.
- byl jsem seznámen s tím, že na moji diplomovou práci se plně vztahuje zákon č. 121/2000 Sb. – autorský zákon, zejména §35 – užití díla v rámci občanských a náboženských obřadů, v rámci školních představení a užití díla školního a §60 – školní dílo.
- беру на вѣдомі, же Высoká škola báňská – Technická univerzita Ostrava (dále jen VŠB-TUO) má právo nevýdělečně ke své vnitřní potřebě diplomovou práci užít (§35 odst. 3).
- souhlasím s tím, že jeden výtisk diplomové práce bude uložen v Ústřední knihovně VŠB-TUO k prezenčnímu nahlédnutí a jeden výtisk bude uložen u vedoucího diplomové práce. Souhlasím s tím, že údaje o diplomové práci, obsažené v Záznamu o závěrečné práci, umístěném v příloze mé diplomové práce, budou zveřejněny v informačním systému VŠB-TUO.
- bylo sjednáno, že s VŠB-TUO, v případě zájmu z její strany, uzavřu licenční smlouvu s oprávněním užít dílo v rozsahu §12 odst. 4 autorského zákona.
- bylo sjednáno, že užít své dílo – diplomovou práci nebo poskytnout licenci k jejímu využití mohu jen se souhlasem VŠB-TUO, která je oprávněna v takovém případě ode mne požadovat přiměřený příspěvek na úhradu nákladů, které byly VŠB-TUO na vytvoření díla vynaloženy (až do jejich skutečné výše).
- беру на вѣдомі, же оdevzdáním své práce souhlasím se zveřejněním své práce podle zákona č. 111/1998 Sb., o vysokých školách a o změně a doplnění dalších zákonů (zákon o vysokých školách), ve znění pozdějších předpisů, bez ohledu na výsledek její obhajoby.

16.5.2014  
V Ostravě : .....



.....

Bc. Ondřej Seidler

Bc. Ondřej Seidler

Výškovice 76

Bílovec, 74301

## ANOTACE DIPLOMOVÉ PRÁCE

SEIDLER, O. *Výukové pracoviště s průmyslovým robotem IRB140: Diplomová práce.* Ostrava: VŠB – Technická univerzita Ostrava, Fakulta strojní, Katedra robototechniky, 2014, 68 s. Vedoucí práce: Novák, P.

Diplomová práce se zabývá tvorbou výukových úloh pro robotizované pracoviště. V úvodu je popsán současný stav zařízení pracoviště na Centru robotiky Katedry robototechniky. V hlavní textové části jsou popsány tři úlohy. Ve všech úlohách je využit článkový dopravník a robot ABB IRB 140. První úloha je zaměřena na manipulaci s kostkami, druhá na třídění dvou různých objektů pomocí systému strojového vidění. Třetí úloha se zabývá pokročilým využitím systému strojového vidění k rozpoznávání dominových kamenů a jejich ukládání na zvolené pozice na vyskladňovací plochu. Všechny úlohy jsou detailně popsány v návodech, které jsou přiloženy v elektronické podobě.

## ANNOTATION OF MASTER THESIS

SEIDLER, O. *Educational Workplace with IRB140 Industrial Robot: Master Thesis.* Ostrava: VŠB VŠB – Technical University of Ostrava, Faculty of Mechanical Engineering, Department of Robotics, 2014, 68 p. Thesis head: Novák, P.

This thesis deals with making of educational tasks for robotic workplace. The introduction contains description of current equipment installed in the Robotics Centre of the Department of Robotics. The main text part contains three tasks. In all tasks is used apron conveyor and ABB IRB 140 robot. First task focus on manipulation with cubes, second on sorting of two different objects using compact vision system. Third task focus on advanced use of compact vision system to identify domino stones and their storing to positions on table. All tasks are described in detail in manuals that are attached in electronic form.

## Obsah

<b>SEZNAM POUŽITÝCH ZNAČEK A SYMBOLŮ .....</b>	<b>8</b>
<b>1 ÚVOD.....</b>	<b>9</b>
<b>1.1 CÍL PRÁCE.....</b>	<b>9</b>
<b>2 ANALÝZA SOUČASNÉHO STAVU PRACOVISTĚ.....</b>	<b>10</b>
<b>2.1 STÁVAJÍCÍ DISPOZICE NA LCR .....</b>	<b>10</b>
<b>2.2 ROBOT ABB IRB 140.....</b>	<b>12</b>
<b>2.3 PERIFERNÍ ZAŘÍZENÍ .....</b>	<b>15</b>
2.3.1 Přísavkový efektor.....	15
2.3.2 Systém rychlé výměny efektorů SWS 005.....	16
2.3.3 Pneumatický rozvaděč.....	17
2.3.4 Výměnný efektor SCHUNK PGN 80/1 .....	19
2.3.5 Výměnný efektor PZN - plus 64 -1 .....	20
2.3.6 Výměnný efektor GWB 64 .....	21
2.3.7 Efektor PGF 80-AS .....	22
<b>2.4 DOPRAVNÍK .....</b>	<b>23</b>
<b>2.5 SYSTÉM STROJOVÉHO VIDĚNÍ.....</b>	<b>25</b>
2.5.1 NI CVS 1454 .....	26
2.5.2 NI Vision I/O Terminal Block .....	27
<b>3 REALIZACE ÚLOH.....</b>	<b>29</b>
<b>3.1 POŽADAVKOVÝ LIST.....</b>	<b>29</b>
<b>3.2 SPUŠTĚNÍ PRACOVISTĚ.....</b>	<b>29</b>
3.2.1 Start ABB IRB 140 .....	29
3.2.2 Start dopravníku .....	30
3.2.3 Start kamerového systému .....	31
<b>3.3 STANOVENÍ RYCHLOSTI DOPRAVNÍKU.....</b>	<b>31</b>
<b>3.4 STAVBA PYRAMIDY.....</b>	<b>33</b>
3.4.1 Programování na FlexPendantu .....	36
3.4.2 Skládání pyramidy.....	36
<b>3.5 TŘÍDĚNÍ DESTIČEK S KRUHOVOU DÍROU .....</b>	<b>40</b>
3.5.1 Program pro roboty .....	42
3.5.2 Program pro CVS .....	45

<b>3.6</b>	<b>ÚLOHA TŘÍZENÍ DOMINOVÝCH KAMENŮ .....</b>	<b>49</b>
3.6.1	Program pro roboty .....	50
3.6.2	Program pro CVS .....	53
3.6.3	Program pro komunikaci přes sériový port .....	57
<b>4</b>	<b>SMART COMPONENT.....</b>	<b>59</b>
<b>4.1</b>	<b>SC DOPRAVNÍK .....</b>	<b>59</b>
<b>4.2</b>	<b>SC EFEKTOR PGN80 PLUS.....</b>	<b>60</b>
<b>4.3</b>	<b>SC OPTICKÁ ZÁVORA .....</b>	<b>62</b>
<b>4.4</b>	<b>PROPOJENÍ SIGNÁLŮ SC.....</b>	<b>62</b>
<b>5</b>	<b>ZÁVĚR .....</b>	<b>63</b>
<b>6</b>	<b>INFORMAČNÍ ZDROJE .....</b>	<b>65</b>
<b>7</b>	<b>SEZNAM PŘÍLOH.....</b>	<b>67</b>

**Seznam použitých značek a symbolů**

<b>Značka</b>	<b>Veličina</b>
clock	Datový typ pro čas
CVS	Compact Vision System - systém strojového vidění
DI	Digital input – digitální vstup
DO	Digital output – digitální výstup
FlexPendant	Dotykový panel pro ovládání robotů
false	Logický datový typ - nepravda
IRC5	Řídicí jednotka robotu
LCR	Laboratoř Centra robotiky
num	Číselný datový typ
OM	Objekt manipulace
pix	Pixel
robtargt	Datový typ pro souřadnice bodu
SC	Smart Component
TCP	Tool Center Point - středový bod nástroje
true	Logický datový typ - pravda
workobject	Datový typ souřadnice pracovního objektu



# 1 Úvod

V posledních letech dochází k rychlému pokroku v oblasti automatizace. Současným trendem automatizace je snížení potřeby přítomnosti člověka při vykonávání určité činnosti. S vývojem řídicích systému se postupně zvyšuje komplexnost automatizace a snižují se požadavky na přítomnost člověka při vykonávání procesu. S narůstající složitostí je ale třeba větší příprava odborníků, kteří jsou schopni realizovat takováto pracoviště.

Pro přípravu studentů pro práci s automatizovanými systémy je k dispozici na Katedře robototechniky výukové Centrum robotiky s modernizovanou laboratoří. V této laboratoři jsou v současné době dvě pracoviště s roboty od firmy ABB, které se využívají pro výukové účely. První pracoviště je vybaveno moderními roboty IRB 140, na druhém pracovišti je aktuálně nejnovější přírůstek – paralelní robot IRB 360. Obě pracoviště jsou přizpůsobeny pro výuku s využitím robotu, dopravníku a systému strojového vidění.

## 1.1 Cíl práce

Cílem práce je připravit pro studenty výukové úlohy, které je seznámí s prací s průmyslovými roboty a dalšími periferiemi, které se nachází na robotizovaném pracovišti. Úlohy jsou vytvořeny s ohledem na možnosti v laboratoři na Centru robotiky.

## **2 Analýza současného stavu pracoviště**

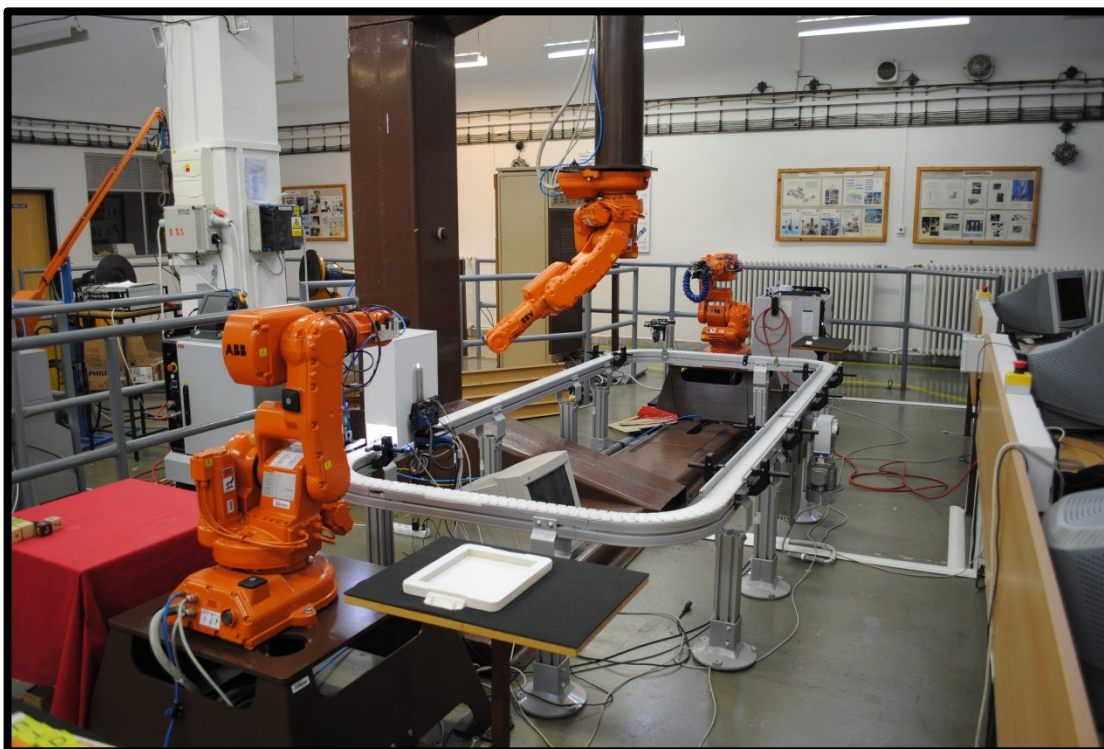
Analýza současného stavu pracoviště se zabývá stručným popisem stavu pracoviště s roboty ABB IRB 140 v laboratoři na Centru robotiky. Zaměřuje se na popis současného stavu robotů, použitelných periférií a vybavení pracoviště.

### **2.1 Stávající dispozice na LCR**

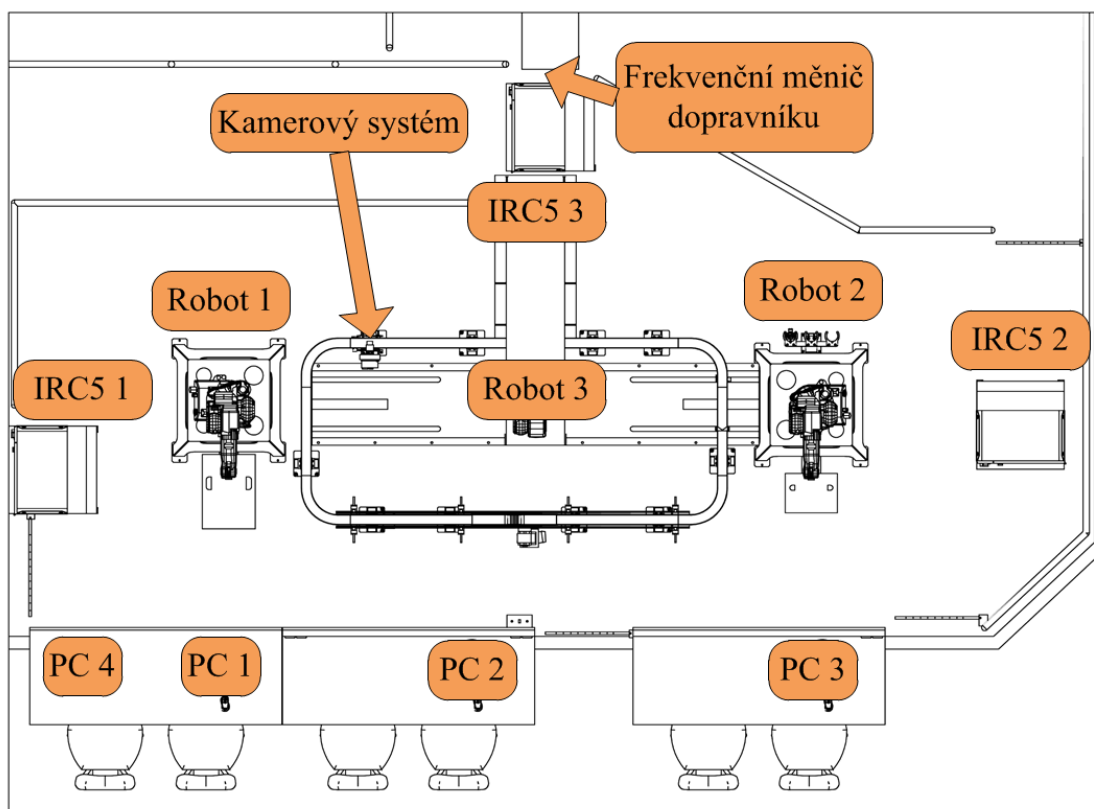
V laboratoři Katedry robototechniky se nachází pracoviště se třemi angulárními šestiosými roboty ABB IRB 140 (Obr. 1). Dva roboty jsou umístěny na stojanech, které jsou stacionárně na podlaze. Třetí robot je zavěšen na sloupové konzoli. Každý robot má svůj vlastní řídicí systém IRC5 s ovládacím panelem FlexPendant pro on-line programování. Řídicí systémy jsou dále připojeny k počítačům, které se nachází na stolech před pracovištěm. Počítače slouží převážně k vytváření simulací s roboty ve virtuálním prostředí pomocí softwaru RobotStudio společnosti ABB. Program k vytvořené simulaci je možné nahrát do řídicí jednotky robotu a odtud ho spouštět či upravovat.

Na robotizovaném pracovišti se nachází několik periferních zařízení. Periferní zařízení tvoří důležitou součást, která určuje charakter pracoviště. Pro přepravu materiálu slouží článkový dopravník Rexroth VFS Direct. V části dopravníku v blízkosti prvního robotu je instalován systém strojového vidění. Se zvyšováním výkonu výpočetní techniky představují kamerové systémy silný nástroj pro analýzu objektů a přinášejí s sebou inteligentní článek při automatizovaném procesu.

První z robotů nese pneumatický přísavkový efektor a druhý je vybaven systémem automatické výměny efektoru. K dispozici má pneumatické efekторы od firmy SCHUNK. Každý z robotů je připojen do sítě se stlačeným vzduchem, aby bylo možné ovládat zmíněné aktivní úchopné hlavice. Pracoviště je ohrazeno bezpečnostními zábranami k zabránění nechtěného vstupu do pracovního prostoru a případné kolizi s roboty. Na zábranách jsou rozmístěna stop tlačítka a mechanické závory.



Obr. 1 Pracoviště ABB IRB 140



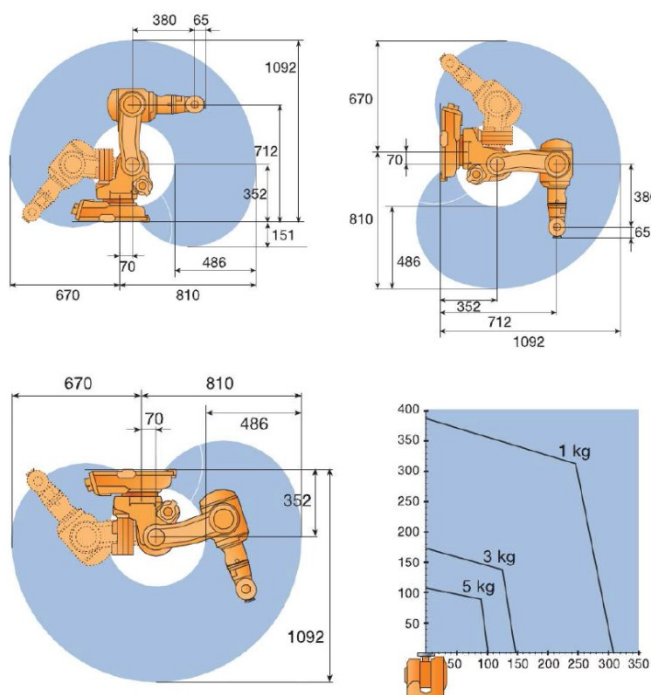
Obr. 2 Schéma rozmístění pracoviště

## 2.2 Robot ABB IRB 140

Tento robot je kompaktní a výkonný šestiosý stroj. Může být namontován na podlahu nebo na zeď v jakémkoli úhlu nebo je možné ho zavěsit. Robustní design je vhodný pro náročná výrobní prostředí. V těle robotu je integrována kabeláž, což usnadňuje jeho zapojení a manipulaci. Robot má funkci detekci kolizí, která přispívá k bezpečnosti při jeho používání. Robot se vyznačuje vysokou střední dobou bezporuchového provozu (MTBF) a malými nároky na údržbu.



Obr. 3 Robot ABB IRB 140



Obr. 4 Pracovní rozsah a zátěžový diagram[10]

Aplikace:	Obloukové svařování	Obsluha strojů
	Montáž	Manipulace s materiálem
	Řezání/Odhrotování	Odebírání z pásu
	Tlakové lití	Balení
	Lepení/Těsnění	Stříkání
	Plastikářské vstřikovací lisy	[10]

**Technická data**

Specifikace:	Nosnost:	6 kg	
	Dosah koncového bodu:	810 mm	
	Doplňující zatížení na koncové rameno:	1 kg	
	Doplňující zatížení na zápěstí:	0,5 kg	
	Integrovaný přívod vzduchu:	8 bar	
Provedení:	Opakovatelnost polohování:	0,03 mm	
	Maximální rychlost:	2,5 m/s	
	Maximální zrychlení:	20m/s <sup>2</sup>	
	Osy:	rozsah	rychlost
	1	360°	200°/s
	2	200°	200°/s
	3	280°	260°/s
	4	400°	360°/s
	5	240°	360°/s
	6	800°	450°/s
El. připojení:	Napětí:	200-600 V, 50/60 Hz	
	Jmenovitý výkon:	4,5 kVA	
	Příkon:	0,4 kW	
Prostředí	Teplota:	5 - 45°C	
	Maximální vlhkost:	95%	

Tab. 1 Parametry [10]

K robotu je připojena řídicí jednotka IRC5. K jednotce je připojen ovládací panel FlexPendant s dotykovým displejem. Jedna IRC5 jednotka dokáže řídit až čtyři roboty, v laboratoři je ale každý robot vybaven vlastní řídicí jednotkou.



*Obr. 5 Řídicí systém s FlexPendantem*

#### Technická data

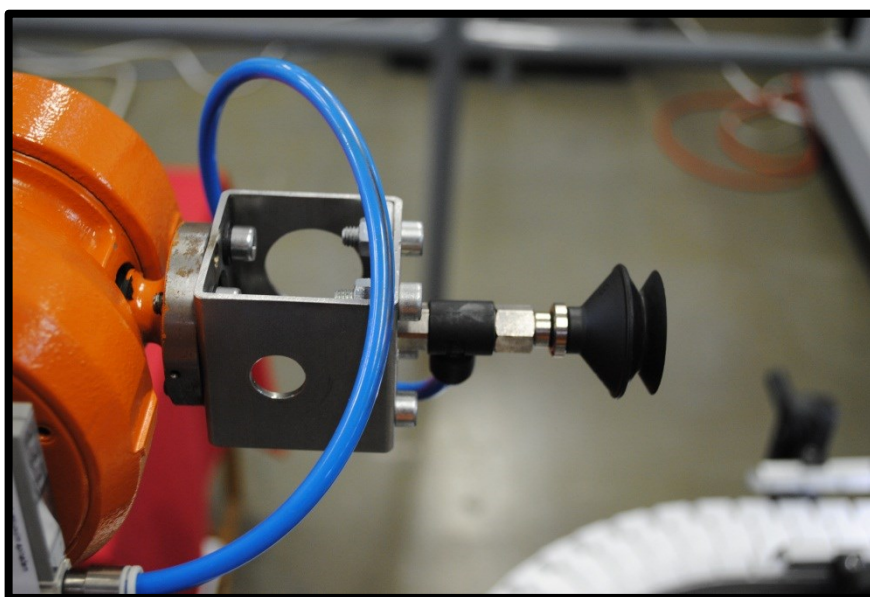
Specifikace:	Hmotnost:	150 kg
	Výška skříně:	970 mm
	Šířka skříně:	725 mm
	Délka skříně:	710 mm
Hardware:	Multi-processor systém, PCI bus, Pentium CPU	
El. připojení:	Napětí:	200-600 V, 50/60 Hz
Provozní podmínky:	Teplota:	0 - 45°C
	Maximální vlhkost:	95%
	Stupeň ochrany:	IP54

*Tab. 2 Parametry řídicího systému*

## 2.3 Periferní zařízení

### 2.3.1 Přísavkový efektor

Efektor s jednou přísavkou se nachází na prvním robotu. Tento efektor je upravenou verzí efektoru, který byl řešen v diplomové práci Ing. Jaroslavem Tomášem. V rámci této diplomové práce byl instalován ejektor ZQ1101U – K1Y5LO – D52CL – Q. Ejektor zajišťuje zpětnou vazbu, díky které lze kontrolovat podtlak potřebný pro uchopení objektu manipulace. Pro správné uchopení OM je potřeba, aby byl povrch hladký, čistý a suchý. [2]



*Obr. 6 Přísavkový efektor*

Ejektor je ovládán pomocí příkazů z řídicí jednotky robotu. Pro ovládání sání slouží digitální výstup DO10\_1 a k ovládání foukání DO10\_2. Obecně lze změnu digitálního signálu pro řídicí jednotku vyjádřit:

SetDO [digitální výstup], [stav signálu (1 | 0)];

Aktivace přísavky:

```
SetDO DO10_1, 1;
```

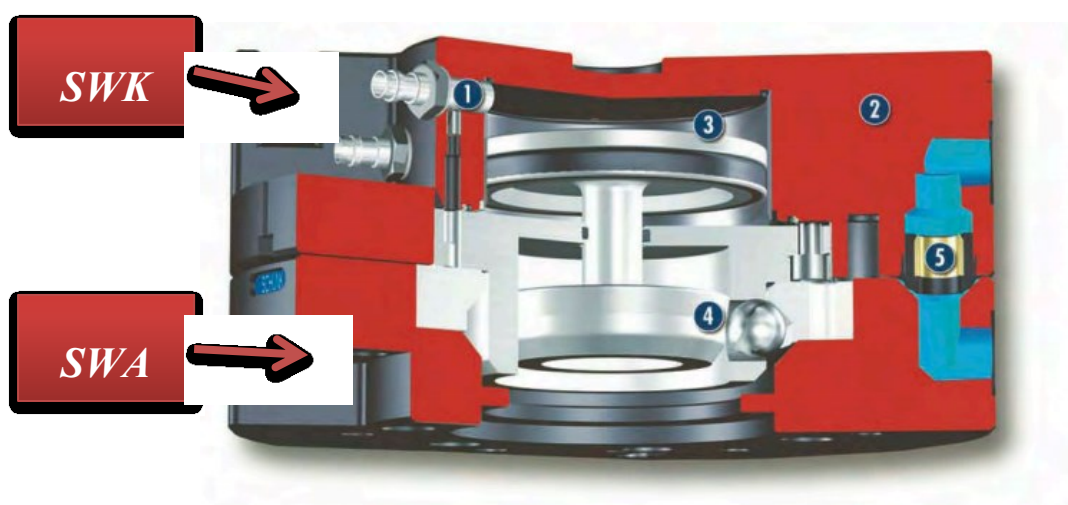
Deaktivace přísavky:

```
SetDO DO10_1, 0;
```



### 2.3.2 Systém rychlé výměny efektorů SWS 005

V praxi nastávají situace, kdy se požaduje od robotu takových operací, při kterých je nutné použít rozdílné efektory. Řešit tento problém lze pomocí systému rychlé výměny efektorů. V laboratoři Centra robotiky je instalován systém SWS 005 firmy SCHUNK. Systém umožňuje automatickou výměnu efektorů ze zásobníku. Do zásobníku, který je v laboratoři, je možno umístit tři efektory (Obr. 8). Realizace systému rychlé výměny je dále popsána v diplomové práci Ing. Jiřího Szczeponiece. [3]

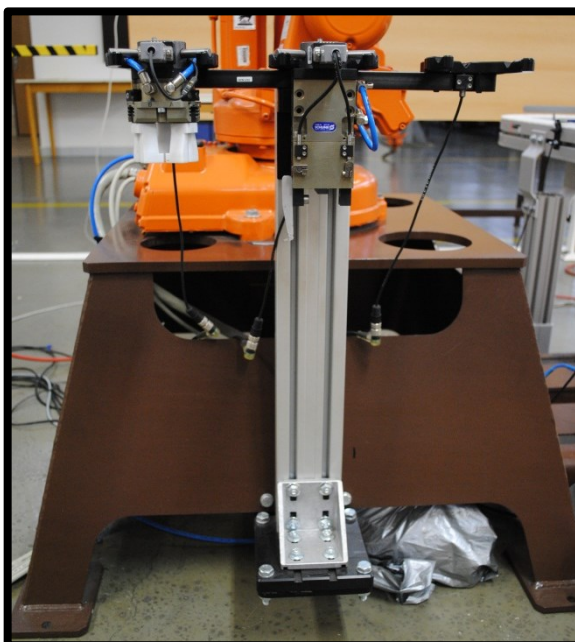


Obr. 7 Systém rychlé výměny SWS 005 [9]

Automatická výměna nástroje robotu zvyšuje flexibilitu robotu. Systém rychlé výměny se skládá z hlavy SWK a adaptéru SWA (Obr. 7). Hlava je napevno připojena k interface robotu. Adaptér je pak připojen ke každému bloku efektoru, u kterého chceme provádět rychlou výměnu.

Pneumaticky poháněný píst (3) tlačí na uzamykací píst (4). Když je píst spuštěn, uzamykací kuličky jsou vtlačeny pod ocelový kroužek a adaptér je vtažen ke hlavě. V případě poklesu tlaku brání třecí síla pístu v pohybu adaptéru. Adaptér může být oddělen od hlavy pouze aktivací pístu. (1) Senzor monitorující uzamčení mechanismu. (5) Pneumatické propojení





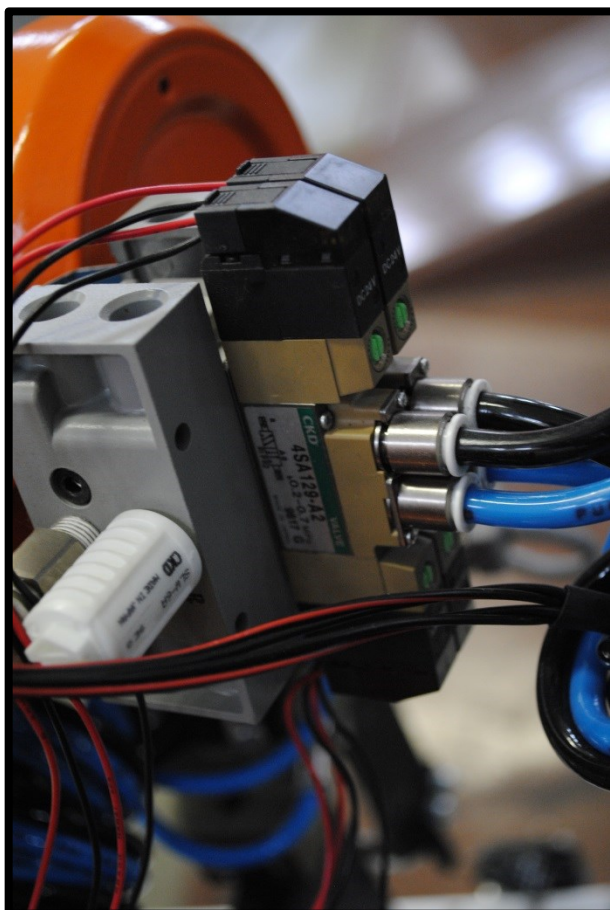
*Obr. 8 Stojan se třemi pozicemi pro efektor*

Dovolené zatížení	8 kg
Statický moment $M_x$ , $M_y$	12,5 Nm
Statický moment $M_z$	17 Nm
Dynamický moment $M_x$ , $M_y$	37,5 Nm
Dynamický moment $M_z$	51 Nm
Maximální odsazení xy při uzamykání	$\pm 1$ mm
Uzamykací síla (6 bar)	710 N
Min/Max mezera uzamčení	0,1,5/3,0 mm
Opakovatelná přesnost	0,01 mm

*Tab. 3 Parametry systému rychlé výměny*

### 2.3.3 Pneumatický rozvaděč

Na robotu se systémem automatické výměny efektoru jsou umístěny dva dvoupolohové, třicestné pneumatické rozvaděče CKD Valve 4SA129-A2. Rozvaděče slouží k ovládání efektorů a k jejich výměně. Pracovní tlak rozvaděčů je 0,2 – 0,7 MPa. Řízení je elektromagnetické se 2 provozními stavy a řídicím napětím 24V. [4]



*Obr. 9 Pneumatické rozvaděče*

Rozvaděč je možno ovládat prostřednictvím digitálních výstupů řídicí jednotky druhého robotu.

Otevření čelistí:

```
SetDO DO10_3,1;
```

Zavření čelistí:

```
SetDO DO10_1,1;
```

Odepnutí efektoru od robotu:

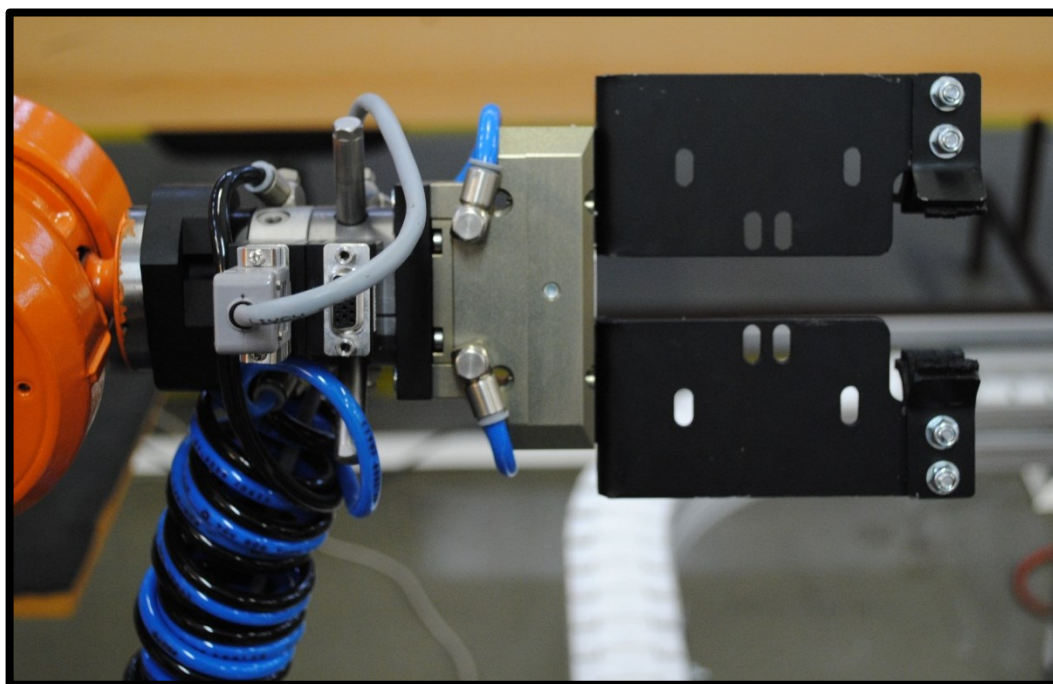
```
SetDO DO10_0,1;
```

Upnutí efektoru k robotu:

```
SetDO DO10_2,1;
```

### 2.3.4 Výměnný efektor SCHUNK PGN 80/1

Efektor PGN 80/1 je aktivní dvouprstý mechanický úchopný systém. Pohon je zajištěn pneumaticky. Z hlediska geometrie je pohyb čelistí lineární. Efektor je standardně dodáván bez úchopných čelistí, které je nutno navrhnout podle parametrů OM. Současné čelisti efektoru jsou vyrobeny z kovu.



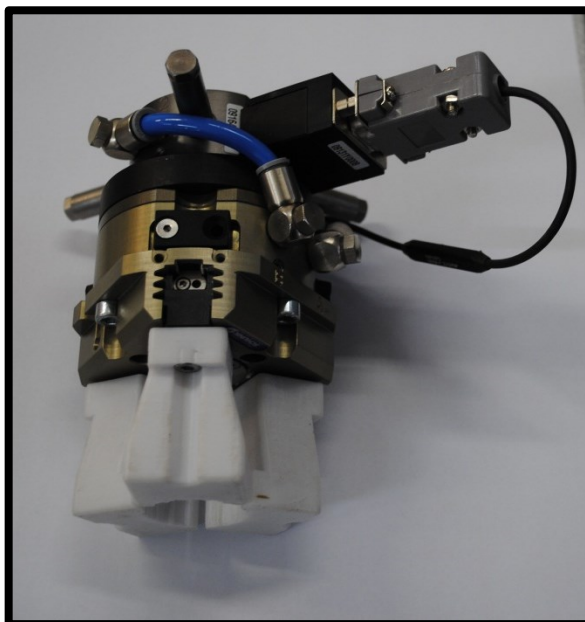
*Obr. 10 Efektor PGN 80/1*

Posuv na čelist	8 mm
Uzavírací síla	360 N
Otevírací síla	400 N
Doporučené zatížení	1,8 kg
Hmotnost	0,43 kg
Min./Max./Nominální tlak	2/8/6 bar
Maximální délka prstu	80 mm
Maximální váha na prst	0,5 kg
Opakovatelná přesnost	0,01 mm

*Tab. 4 Efektor PGN 80/1[8]*

### 2.3.5 Výměnný efektor PZN - plus 64 -1

Tento tříprstý efektor společnosti SCHUNK se vyznačuje velkými úchopnými silami a krouticím momentem. Koncové čelisti byly vyrobeny z Rapid Prototypingu a jsou popsány v diplomové práci Ing. Jiřího Szczeponiece. [3]



*Obr. 11 Efektor PZN – plus 64-1*

Píst se pohybuje nahoru a dolů pomocí stlačeného vzduchu a tlačí na zkosené aktivní plochy čelistí. Takto dochází k transformaci pohybu pístu rovnoměrně na všechny tři čelisti.

Posuv na čelist	6 mm
Uzavírací síla	580 N
Otevírací síla	640 N
Doporučené zatížení	2,9 kg
Hmotnost	0,43 kg
Min./Max./Nominální tlak	2/8/6 bar
Maximální délka prstu	90 mm
Maximální váha na prst	0,35 kg
Opakovatelná přesnost	0,01 mm

*Tab. 5 Technické parametry PZN – plus 64-1[12]*

### 2.3.6 Výměnný efektor GWB 64

Jde o dvoučelistový úhlový pneumatický efektor s úhlovými čelistmi. Jeho využití je vhodné v aplikacích, kde jsou kladeny velké požadavky na uchopovací síly a zároveň je třeba velmi rychlého sevření pomocí úhlových čelistí.



Obr. 12 Efektor GWB 64

Otevírací úhel na čelist	90°
Uzavírací moment	27,45 Nm
Doporučené zatížení	2,2 kg
Hmotnost	0,85 kg
Min./Max./Nominální tlak	4/6,5/6 bar
Maximální délka prstu	80 mm
Maximální váha na prst	0,32 kg
Opakovatelná přesnost	0,1 mm

Tab. 6 Technické parametry GWB 64 [7]

### 2.3.7 Efektor PGF 80-AS

Jedná se o paralelní pneumatický efektor se dvěma čelistmi. Díky dlouhému vedení čelistí je efektor odolný během operací, kdy vzniká na uchopovacích čelistech vysoký moment.



*Obr. 13 Efektor PGF 80-AS*

Posuv na čelist	16,5 mm
Uzavírací síla	760 N
Minimální síla přes úchopnou pružinu	200 N
Doporučené zatížení	2,8 kg
Hmotnost	1,25 kg
Min./Max./Nominální tlak	3,5/8/6 bar
Maximální délka prstu	90 mm
Maximální váha na prst	0,75 kg
Opakovatelná přesnost	0,02 mm

*Tab. 7 Technické parametry efektoru PGF 80-AS [20]*



## 2.4 Dopravník

Součástí robotizovaného pracoviště tvoří článkový dopravník Rexroth VFS Direct s elektrickým pohonem a frekvenčním měničem Mitsubishi FR-D 720S-042. Článkový pás tvoří nekonečnou smyčku složenou s plastových segmentů. Dopravník byl realizován v diplomové práci Ing. Kamila Byrtuse, kde jsou uvedeny technické údaje k dopravníku. Z hlediska řízení je frekvenční měnič připojen kabelem k řídicí jednotce IRC5 prvního robotu. [1]



*Obr. 14 Článkový dopravník Rexroth VFS Direct*

Dopravník není vybaven snímačem jeho aktuální polohy a je tedy nutné zjišťovat polohu OM jiným způsobem. V současné době jsou na dopravníku optické závory, které snímají přítomnost OM. V aktuálním provedení je dopravník možno spustit ve třech rychlostech příslušným příkazem v řídicí jednotce IRC5 prvního robotu. Článkový dopravník se může pohybovat pouze v jednom směru, jinak hrozí poškození plastových článků.

Ovládání dopravníku přes IRC5:

Před ovládáním rychlosti dopravníku je třeba povolit běh dopravníku:

```
SetDO DO10_3,1;
```

Po povolení běhu dopravníku je možno spustit pohyb zvolenou rychlostí:

```
SetDO DO10_4,1;    !nízká rychlost
SetDO DO10_5,1;    !střední rychlost
SetDO DO10_6,1;    !vysoká rychlost
```

Pro ukončení pohybu dopravníku je nutné změnit signál příslušné rychlosti na nulu a zakázat běh dopravníku. Zastavení pro vysokou rychlost:

```
SetDO DO10_6,0;
SetDO DO10_3,0;
```

Při případné změně rychlosti dopravníku na jinou rychlost je nejprve třeba změnit signál aktivní rychlosti a následně signál požadované nové rychlosti. Pro změnu z vysoké rychlosti na střední:

```
SetDO DO10_6,0;    !deaktivace vysoké rychlosti
SetDO DO10_5,1;    !střední rychlost
```

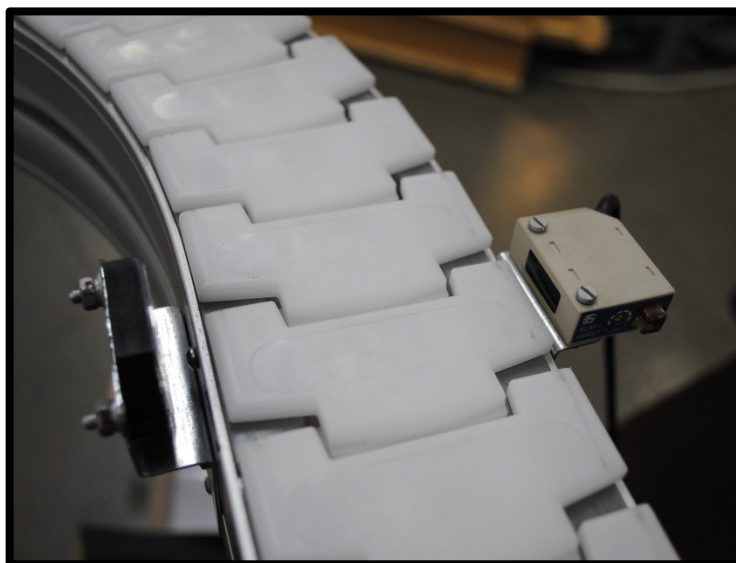
Na dopravníku jsou umístěny dvě optické závory Telemecanique XUM-LH0451 (Obr. 15). Pomocí závor lze snímat průchody objektů na dopravníku. První závora je umístěna u systému strojového vidění a je propojena se systémem strojového vidění a řídicí jednotkou prvního robotu (Obr. 16). Digitální výstup druhé závory je přiveden na řídicí jednotku druhého robotu.

#### Technické parametry

Jmenovité napětí [V]:	12-24 DC
Rozsah napětí [V]:	10-30 DC
Max. spínaný proud [mA]:	100
Maximální spínací frekvence [Hz]:	500
Zpoždění [ms]:	1

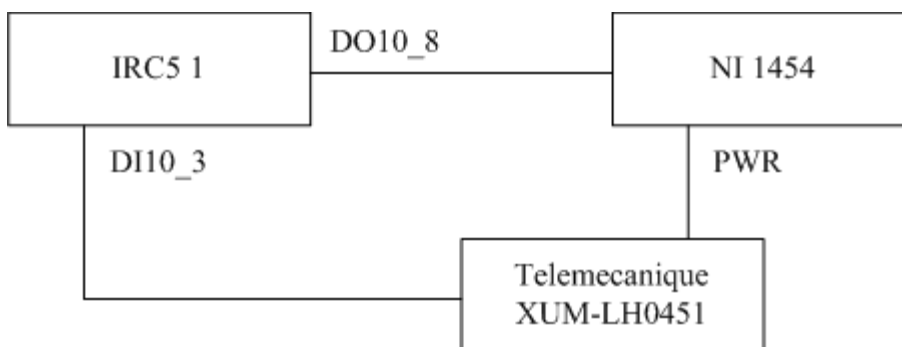
Tab. 8 Parametry optické závory [21]





*Obr. 15 Telemecanique XUM-LH0451*

Pokud dojde k průchodu objektu přes první závoru, je vyslán signál DI10\_3 do řídicí jednotky robotu. V probíhajícímu programu řídicí jednotky je možné nastavit reakci a vyslat signál DO10\_8 pro systém strojového vidění.



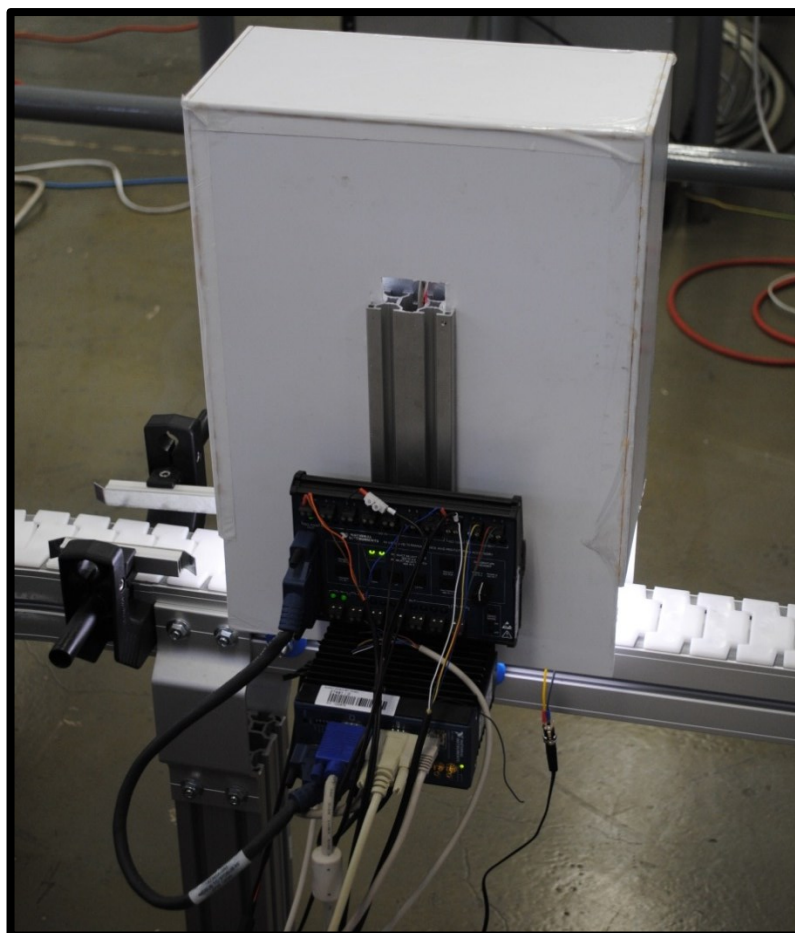
*Obr. 16 Schéma propojení signálů s prvním robotem*

Druhá optická závora je připojena na digitální vstup DI10\_6 druhého robotu.

## 2.5 Systém strojového vidění

Systém strojového je tvořen kamerou firmy Basler s objektivem Fujinon HF16HA-1B, která je připojena k vision systému National Instruments CVS-1454 s rozšířením NI Vision I/O Terminal Block. Dopravník je v oblasti kamery nasvětlen LED páskou, která má

vlastní napájecí zdroj. Osvětlení je zakrytováno boxem z PVC k zajištění vhodných světelných podmínek. Úprava osvětlení dopravníku a realizace krycího boxu je popsána v diplomové práci Ing. Tomáše Chamrada. [4]



*Obr. 17 Systém strojového vidění*

### **2.5.1 NI CVS 1454**

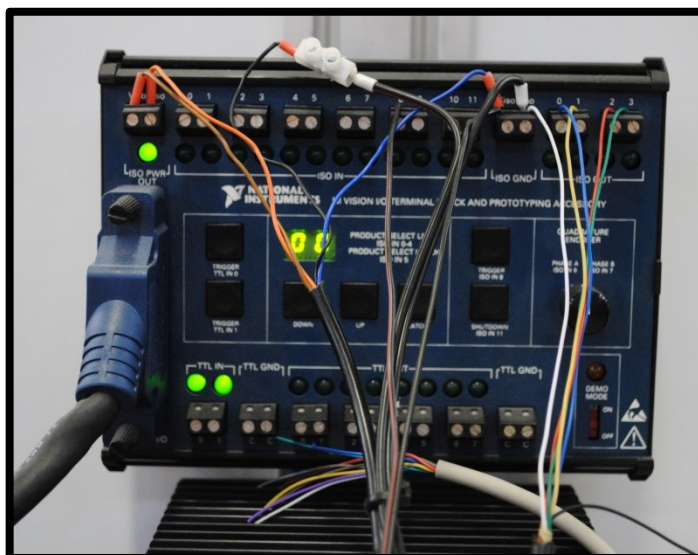
National Instruments CVS 1454 představuje kompaktní kamerový systém pro identifikaci, zpracování a analýzu obrazu. Kompaktní systém je vhodný pro nasazení do nepříznivých podmínek běžných v robotizovaných buňkách. Jednotka je vybavena třemi sběrnicemi IEEE 1394, které umožňují připojení až tří kamer. Tato vlastnost je příznivá v aplikacích, kde dochází k několika kontrolám pomocí strojového vidění, protože není potřeba pro každou kameru vlastní řídicí systém. [13]



Obr. 18 NI CVS 1454

### 2.5.2 NI Vision I/O Terminal Block

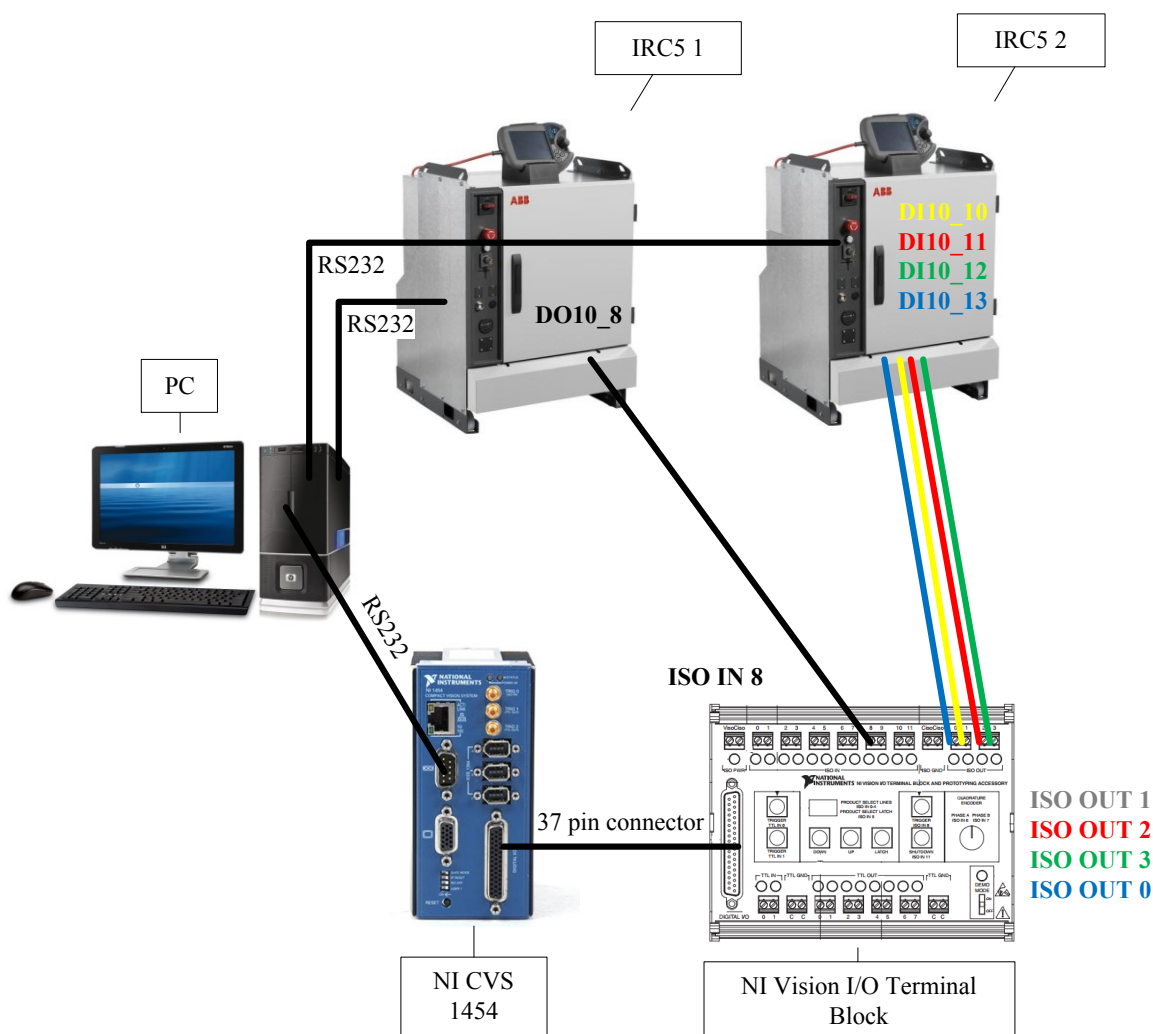
NI Vision I/O Terminal Block je rozšiřující karta systému NI CVS 1454. Karta je připojena k CVS pomocí 37 pinového konektoru. Na kartě se nacházejí digitální vstupy a výstupy, které rozšiřují možnosti práce s kamerovým systémem. [14]



Obr. 19 NI Vision I/O Terminal Block

Pro zvýšení efektivity a rychlosti programování inspekčních úloh je možné použít dostupný software NI Vision Builder for Automated Inspection. Pomocí tohoto programu lze zachytit a zpracovat snímek ze systému strojového vidění. Program na základě výsledků inspekce rozhoduje, zda zkoumaný objekt vyhověl naprogramovaným

podmínkám. Komunikace může být realizována pomocí modulu DI/O, sériového portu nebo ethernetu.



Obr. 20 Blokové schéma systému strojového vidění

### 3 Realizace úloh

Úlohy jsou přizpůsobeny možnostem na pracovišti IRB 140. Použití robotu je závislé na druhu jeho efektoru. V případě potřeby periferií ve formě fixačních přípravků, čelistí atp. je nutné pracoviště o tyto komponenty doplnit.

#### 3.1 Požadavkový list

- Požít dostupné roboty ABB IRB 140.
- Použít stávající článkový dopravník.
- Navrhnout robotizované pracoviště se sníženou úrovní prašnosti a hlučnosti.
- Výuková aplikace by měla být navržena pro prostory v laboratoři katedry robototechniky.
- Opakovatelné použití výukové pomůcky.
- Maximální nosnost robotu 6 kg.
- V navržené aplikaci použít úchopové hlavice společnosti SCHUNK Intec s.r.o., které jsou k dispozici na stanoveném pracovišti.
- Dle nutnosti navrhnout případná periferní zařízení.
- Sepsat detailní návod pro řešené úlohy

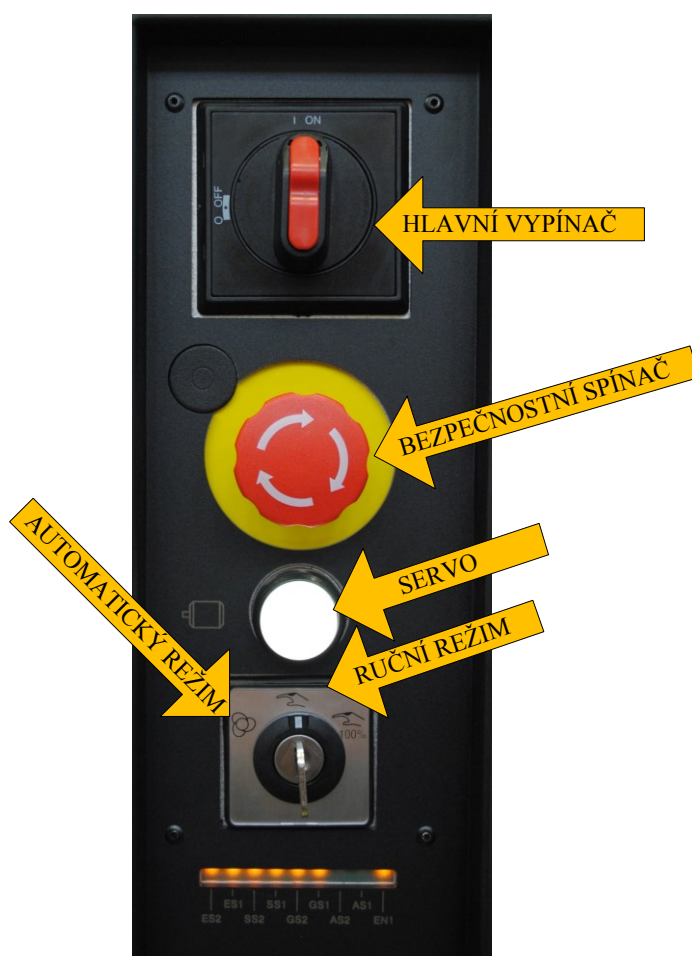
#### 3.2 Spuštění pracoviště

Pracoviště s roboty IRB 140 zahrnuje řadu periferií, které byly popsány v kapitole 2. Ke zdárnému chodu pracoviště je třeba skontrolovat, zda jsou potřebná zařízení zapojena do elektrické sítě a stav hlavních vypínačů. Následující řádky popisují operace vedoucí k bezproblémovému chodu vybavení.

##### 3.2.1 Start ABB IRB 140

1. Spustit vypínač hlavního napájení na panelu za PC 2 (Obr. 2). Pokud je aktivní, svítí zelená kontrolka.
2. Přepnout hlavní vypínač na panelu řídicí jednotky do polohy „ON“. Na FlexPendantu se zobrazí načítací obrazovka.

3. Kontrola bezpečnostních spínačů (FlexPendant, spínače závor, spínače na bezpečnostním hrazení). V případě rozepnutí některého ze spínačů se na displeji FlexPendantu objeví chybové hlášení.
4. V případě výpadku napájení robotu ze záložní baterie je nutné provést kalibraci.
5. Pro práci v ručním režimu nastavit přepínač pomocí klíče do polohy s ikonou ruky.



Obr. 21 Hlavní ovládací panel řídicí jednotky IRC5

### 3.2.2 Start dopravníku

1. Frekvenční měnič dopravníku je umístěn na sloupu vedle řídicí jednotky třetího robotu (Obr. 2).
2. Hlavní vypínač přesunout do polohy „1“. Pokud nedojde ke spuštění frekvenčního měniče (nerozsvícený displej), zkontrolovat polohu pojistky ve skříni.

3. Ovládání chodu je možné regulovat přes řídicí jednotku prvního robotu.



*Obr. 22 Frekvenční měnič dopravníku*

### 3.2.3 Start kamerového systému

1. Zasunout zástrčku kamerového systému a LED osvětlení do síťové zásuvky 230 V.
2. Pro tvorbu programu spustit v PC 4 aplikaci NI Vision Builder for Automated Inspection.
3. Při zapnutí kamerového systému se objeví v nabídce aplikace připojení k CVS. Bližší popis k tvorbě programu je v příloze C.

### 3.3 Stanovení rychlosti dopravníku

Pro odebrání předmětů za chodu dopravníku je nutné znát jeho rychlost. Z dostupných údajů k dopravníku nebylo možné tuto informaci zjistit, proto jsem se uchýlil k experimentálnímu měření rychlosti. Vyloučil jsem možnost měření času pomocí obyčejných stopek ovládaných člověkem, který by stopoval čas průchodu objektu na stanovené dráze, protože tento způsob by přinesl z důvodu lidského faktoru nežádoucí nepřesnost.



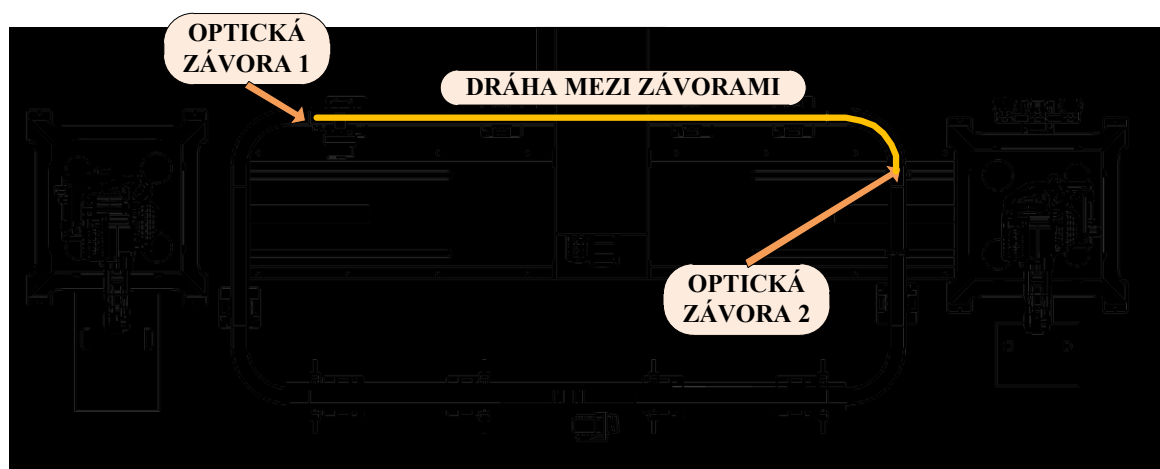
Optimální možností pro realizaci úkolu je použití optických závor, protože z rozhodování o přítomnosti objektu v daném místě se vyloučí lidský faktor. Řídící jednotky robotů umožňují měřit čas a ukládat ho do datového typu „*clock*“. V řídicí jednotce byl sestaven program, který funguje jako stopky. [15]

```

VAR clock cas;           !definice proměnné „cas“ typu clock
VAR num casNum:=0;       !definice proměnné typu num
PROC Stopky()
    ClkReset cas;         !resetuje čas na nulu
    WaitDI DI10_13,1;     !signál z první závory
    ClkStart cas;         !start měření času
    WaitDI DI10_6,1;      !signál z druhé závory
    ClkStop cas;          !stop měření času
    casNum := ClkRead(cas); !převod clock na num
    TPWrite ""\Num:=casNum; !výpis hodnoty na FlexPendant
ENDPROC

```

V programu je nejprve nadefinována proměnná „cas“ typu *clock*, která ukládá čas a proměnná „casNum“ typu *num*, která se používá pro vyjádření číselné hodnoty. Program na začátku resetuje hodnotu času na nulu a poté čeká na signál z první optické závory. Když dostane signál, odstartuje se počítání času a to až do doby, kdy systém dostane signál s druhé optické závory a ukončí měření času. Časovou proměnnou je nutné převést na typ *num*, protože pro výpis na FlexPendantu není možné použít datový typ *clock*.



Obr. 23 Znázornění experimentu



Dráha mezi závorami byla změřena na modelu pracoviště v softwaru Creo a následně byla vzdálenost ověřena reálným měřením v laboratoři. Vzdálenost byla určena na **3002,70 mm**. Výsledky měření shrnuje Tab. 9.

Rychlost	Naměřený čas [s]	Rychlost [mm/s]
Nízká	70,96	42,32
Střední	23,34	128,65
Vysoká	13,90	216,02

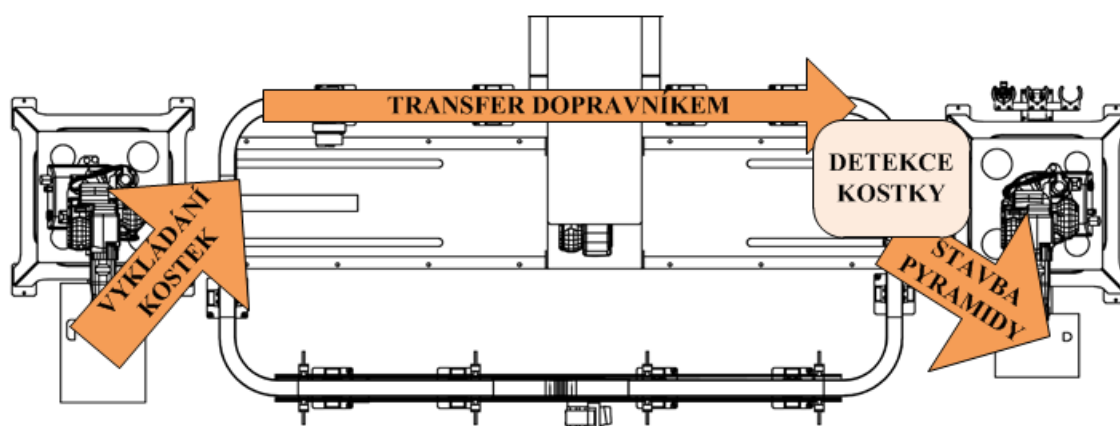
*Tab. 9 Rychlosti dopravníku*

### 3.4 Stavba pyramidy

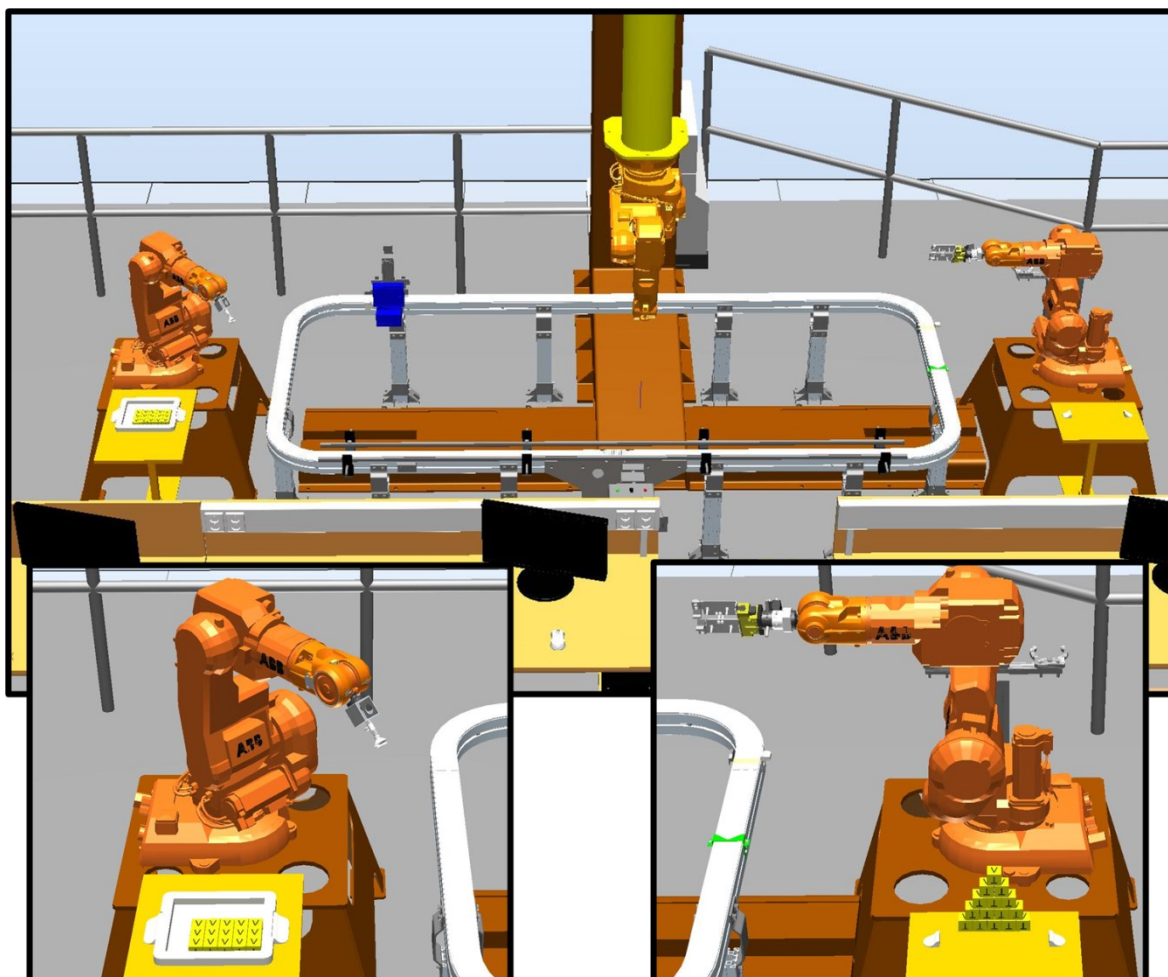
Tato úloha je určena pro seznámení se základními ovládacími prvky na pracovišti. První robot má za úkol vykládat patnáct dřevěných kostek na pohybující se dopravník. Kostky jsou umístěny v zásobníku na odkládacím stole. K zvládnutí tohoto procesu je třeba umět ovládat přísavku a běh dopravníku. Druhý robot čeká v pozici a po projetí kostky optickou závorou, dostává impuls pro pokračování v programu a odebírá kostky. Kostky jsou ukládány na odkládací stůl poblíž robotu a to v takovém uspořádání, že po celkovém vyložení kostek vytvoří pyramidu. (Obr. 25)

Úloha bude rozdělena na dvě části z hlediska přístupu k programování daných úloh. První část má za úkol seznámit studenta s programováním prostřednictvím FlexPendantu (dále FP). V prostředí dotykového panelu FP se student naučí založit nový program a zvládne vytvářet základní pohybové instrukce. Úspěšným dokončením úlohy bude kompletní program, jehož prostřednictvím bude robot přemísťovat kostky ze zásobníku do vhodné pozice na spuštěný dopravník.

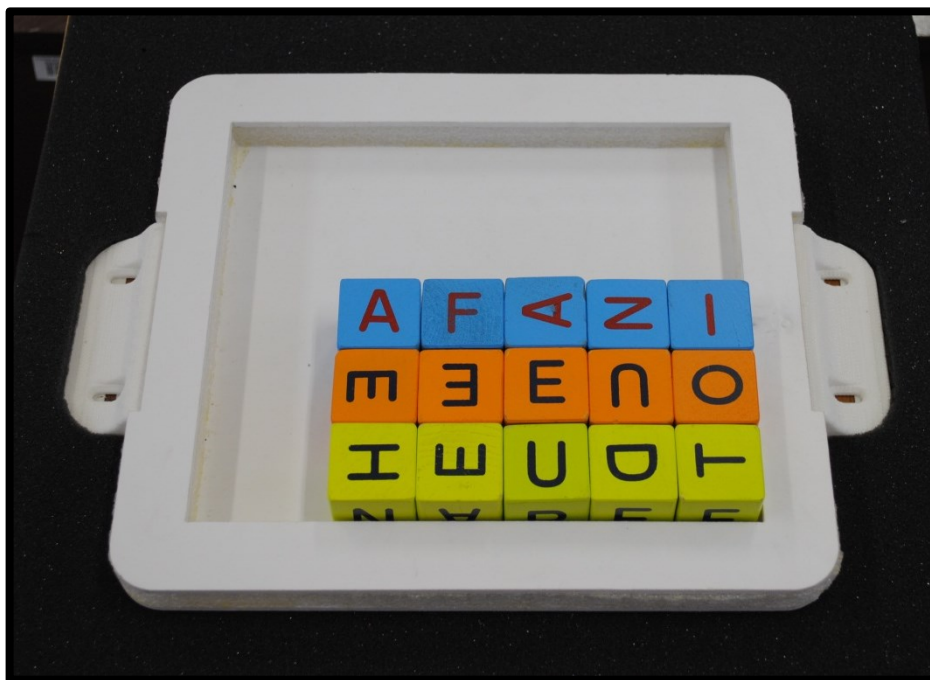
Cílem druhé části úlohy bude vyložení kostek do tvaru pyramidy. Kostky bude nutné odebírat za pohybu dopravníku. Situace, kdy je třeba manipulovat s předměty za pohybu, nastává v praxi často. Důvodem pro tento typ zacházení s objekty manipulace je požadavek na zkrácení pracovního cyklu robotu a důraz na spojitý pohyb dopravníku. Vhodný přístup k řešení problému se stává z vytvoření simulace v RobotStudiosu a následné nahrání programu do řídicí jednotky robotu. Student se seznámí s problematikou tvorby simulace a s obtížemi, které vznikají při přenášení programu do reálné aplikace.



Obr. 24 Tok materiálu při stavbě pyramidy



Obr. 25 Pohled na pracoviště v RobotStudiu, vykládání ze zásobníku (vlevo dole), stavba pyramidy (vpravo dole)



*Obr. 26 Zdrojové kostky v zásobníku u prvního robotu*



*Obr. 27 Vytvořená pyramida*

### 3.4.1 Programování na FlexPendantu

#### **V úloze se bude řešit:**

- práce s dotykovým ovládáním,
- jak bezpečně pohybovat s robotem,
- práce s proměnnými.

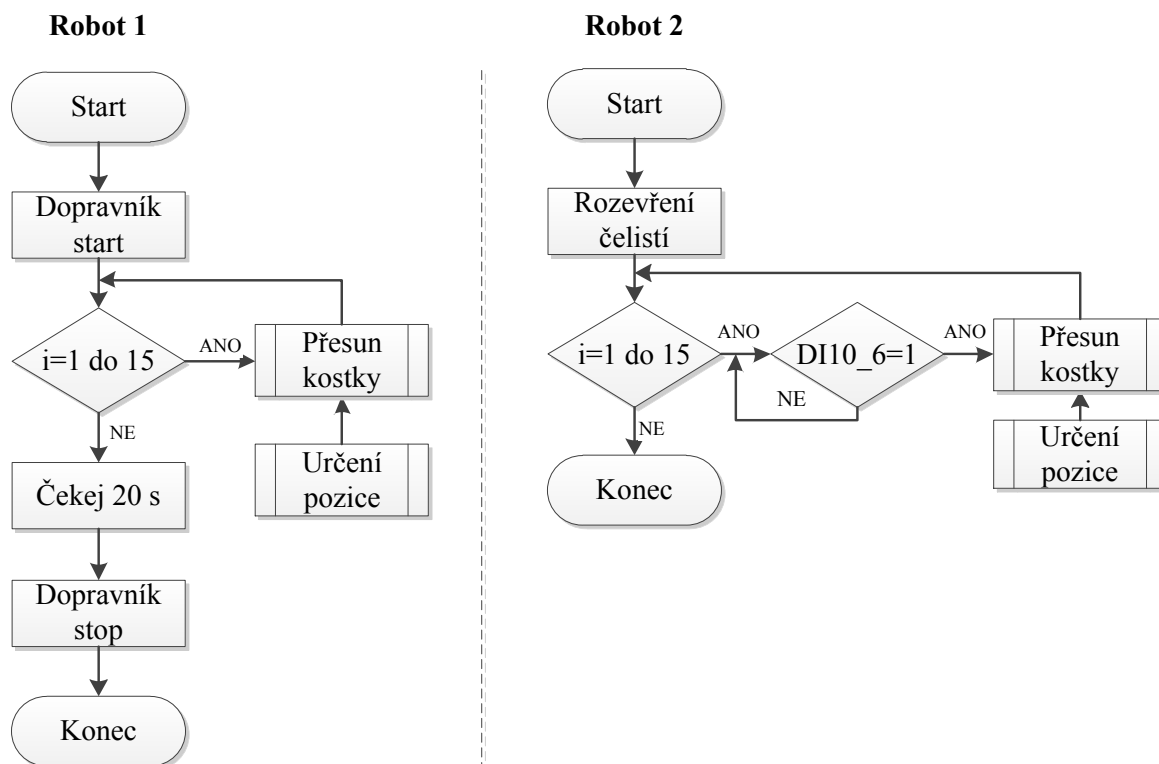
Vykládání kostek na dopravník představuje vhodnou úlohu pro programování pomocí FlexPendantu. V příloze A je popsán podrobný návod vedoucí k vytvoření programu. Zkušenosti s programováním robotů ve výuce mě vedou k názoru, že studenti by vytvořili pro každou kostku jedinečný bod a pohybové instrukce by přidávali za sebe až do kompletního vyložení. Program napsaný tímto způsobem je nepřehledný a špatně upravitelný. Pro online programování na FP jsem zvolil algoritmus používající proměnný robtarget a cyklus, který je použit i v simulaci kompletní úlohy.

### 3.4.2 Skládání pyramidy

#### **V úloze se bude řešit:**

- tvorba logiky v simulaci,
- použití funkce Smart Component,
- tvorba workobjectů a robtargetů,
- upload programu do reálné řídicí jednotky.

Na Obr. 28 je znázorněn vývojový diagram úlohy pro jednotlivé roboty. Programy běží nezávisle na sobě v řídicích jednotkách odpovídajících robotů. Právě z důvodu nezávislého chodu je v programu prvního robotu časová prodleva dvacet sekund před zastavením dopravníku. Tento čas je dostatečně dlouhý pro dopravení poslední manipulované kostky k druhému robotu. Detailní popis úlohy je v příloze B.



Obr. 28 Vývojový diagram stavby pyramidy

Pro vykládání kostek prvním robotem je třeba patnáct pozic, kdy plocha přísavky přiléhá ke stěně kostky. Je vhodné se ke kostce přibližovat shora, protože při přiblížení přísavky z boku by mohlo dojít k posunutí kostky. Když je kostka uchycená a robot ji vytahuje ze zásobníku, situace již vyžaduje vytáhnutí kostky ve směru vertikálním, jinak by docházelo ke kolizím. Počet pozic pro bezpečnou manipulaci se tedy zdvojnásobí. Bylo by nepraktické nastavovat zvlášť třicet samostatných pozic, když se kostky nacházejí v pravidelných rozestupech od sebe. Programovací jazyk RAPID umožňuje tento problém řešit pomocí proměnného robtargetu. V dalších řádcích je zobrazena část kódu se stěžejními funkcemi použitými při řešení úlohy. [15]

```

!souradnice na první pozici VAR robtaret ZasobnikFine:=[[x,y,z],...
VAR robtaret
ZasobnikFine:=[[0,0,0],[1,0,0,0],[0,0,0,0],[9E9,9E9,9E9,9E9,9E9,9E9]];
...
!vypocet souradnic pro dalsi cyklus
x:=x+35;           !posun souradnice v ramci aktualni rady
if i=5 or i=10 then
    y:=y+35;       !posun rady

```

```

        x:=0;           !pozice první kostky v rade
    endif

    ...

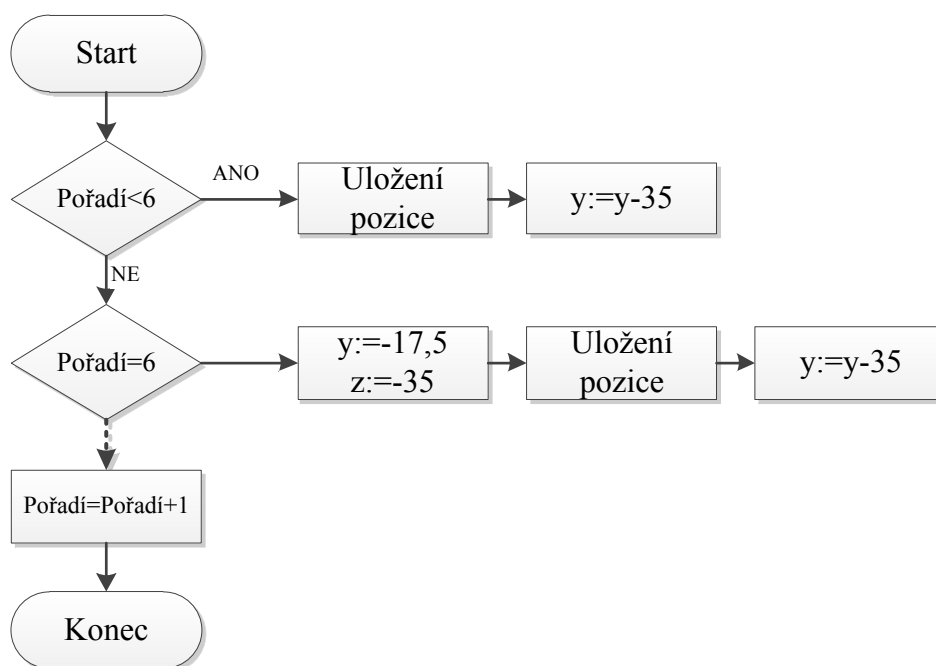
    !prirazení souřadnic x, y do robtargetu
    ZasobnikFine.trans.x:=x;
    ZasobnikFine.trans.y:=y;

    ...

```

Souřadnice bodu v prostoru se ukládají v první hranaté závorce při definování robtargetu ve tvaru [x,y,z]. V každém cyklu probíhá výpočet souřadnic pro další cyklus. Hrana kostky měří 35 mm. V podmínce stačí tuto délku přičítat k aktuální hodnotě proměnné x, y. Tuto hodnotu přiřadíme souřadnici robtargetu tím způsobem, že za název bodu, který editujeme, napíšeme „**trans.x:=**“ a přiřadíme hodnotu pro osu x.

Při odebírání kostky z dopravníku druhým robotem a ukládání do pyramidy jsem využil stejného principu s proměnným robtargetem. Pro lepší přehlednost jsem pro výpočet a ukládání souřadnic vytvořil vlastní rutinu. Obr. 29 zobrazuje diagram pro prvních 6 kostek v pořadí. Pro další kostky je princip změny souřadnic obdobný.



Obr. 29 Vývojový diagram pro tvorbu robtargetů pyramidy

Pro odebírání kostek za pohybu jsem využil rychlost zjištěnou z experimentu. V programu druhého robotu jsem definoval rychlost pomocí Speed data. Stejným způsobem jsou nadefinovány defaultní rychlosti v systémovém modulu BASE. [15]



**Rychlosti**

Name	TCP speed	Orientation	Linear ext. axis	Rotating e. axis
v5	5 mm/s	500°/s	5000 mm/s	1000°/s
v10	10 mm/s	500°/s	5000 mm/s	1000°/s

*Tab. 10 Příklad definovaných rychlostí [15]*

```
!dopravnik na nejvyšší rychlost 216 mm/s
const speeddata vDopravnik:=[ 216, 30, 200, 15 ];
```

- 216 v hranaté závorce udává rychlost TCP (tool centre point) v mm/s.
- 30 °/s pro orientaci efektoru.
- 200 mm/s pro lineární externí osy.
- 15 °/s pro rotaci externích os.

K uchopení kostky jsem použil trigger, který slouží ke spouštění událostí během pohybu robotu. Tato vlastnost umožní zavřít čelisti ještě během pohybu robotu z bodu do bodu.

```
var triggdata zavri;

! spusti DO10_1 0.08 s před dosažením souřadnice bodu OdberKostky
TriggIO zavri, 0.08\Time \DOp:=DO10_1, 1;
TriggL\Conc, OdberKostky,vDopravnik,zavri,fine,PGN80Plus_1\WObj:=Odber;
```

TriggIO lze nastavit buď vzdálenost, nebo v mém případě vhodnější čas (0,08 s). Pokud se robot nachází v definované vzdálenosti/čase před dosažením bodu změní se hodnota signálu DO10\_1, která slouží pro zavření efektoru. V datasheetu k efektoru PGN 80/1 [8] uvádějí dobu k přestavení mechanismu na 0,04 s. Dvojnásobný čas byl zvolen s ohledem k bezpečnosti uchopení. Instrukce TriggL je téměř shodná s pohybovou instrukcí MoveL. Rozdíl představuje zahrnutí triggerů pro spuštění události.

### 3.5 Třídění destiček s kruhovou dírou

V této úloze probíhá manipulace se dvěma destičkami (Obr. 30) s obdélníkovým půdorysem. Destičky se liší velikostí průměru vyvrtané díry kruhového průřezu. První robot má za úkol vykládat destičky ze zásobníku na odkládacím stole na pohybující se dopravník podobně jako v předchozí úloze. Jako materiál pro výrobu jsem využil desku z PVC o tloušťce 4 mm, která zůstala v laboratoři jako odpadní materiál z realizace světelného boxu pro kameru. Pro testování byla plocha destiček zvýrazněna modrým lihovým fixem a po zdařilém odzkoušení bylo zvýraznění nahrazeno vrstvou tvrdého papíru modré barvy. Pro slepení bylo použito lepidlo vhodné pro lepení plastů. Průměr vyvrtaných děr je 5 mm a 8 mm.



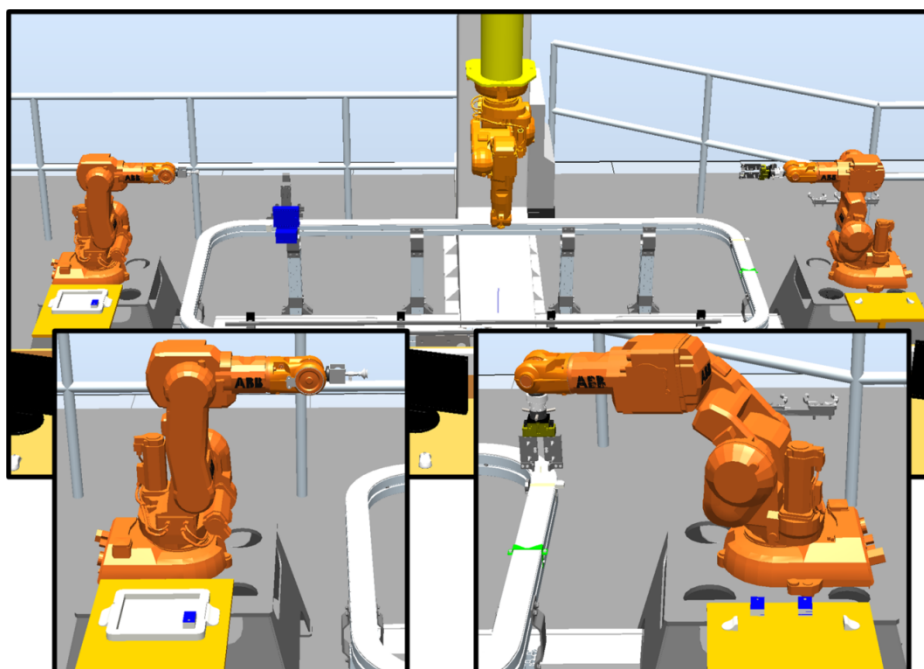
*Obr. 30 Destičky s předvrtanou dírou*

Prostřednictvím kamery se hranoly rozlišují na dva typy podle velikosti díry. Informace je odeslána pomocí rozšiřujícího modulu NI Vision I/O Terminal Block do řídicí jednotky druhého robotu. Druhý robot čeká v pozici nad optickou závorou a po projetí destičky, dostává řídicí jednotka impuls, který odstartuje odebrání. Destičky jsou ukládány na odkládací stůl poblíž robotu a to do dvou sloupců. V každém sloupci jsou hranoly se shodným průměrem díry. Při tvorbě programu je třeba vzít v úvahu, že destičky přicházejí po dopravníku v nahodilých kombinacích. Odebírání objektu manipulace probíhá za chodu dopravníku. Při ukládání do sloupců je třeba zvolit polohu pro ukládání podle počtu uložených kostek z předchozích kroků.

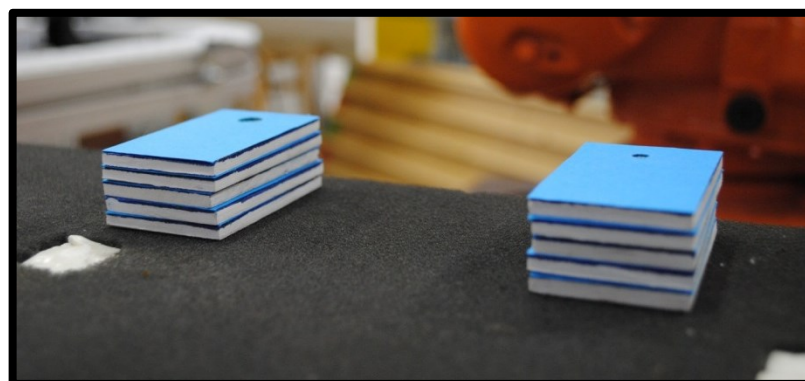




Obr. 31 Tok materiálu při třídění hranolů



Obr. 32 Pohled na pracoviště v RobotStudiu, vykládání ze zásobníku (vlevo dole), třídění (vpravo dole)



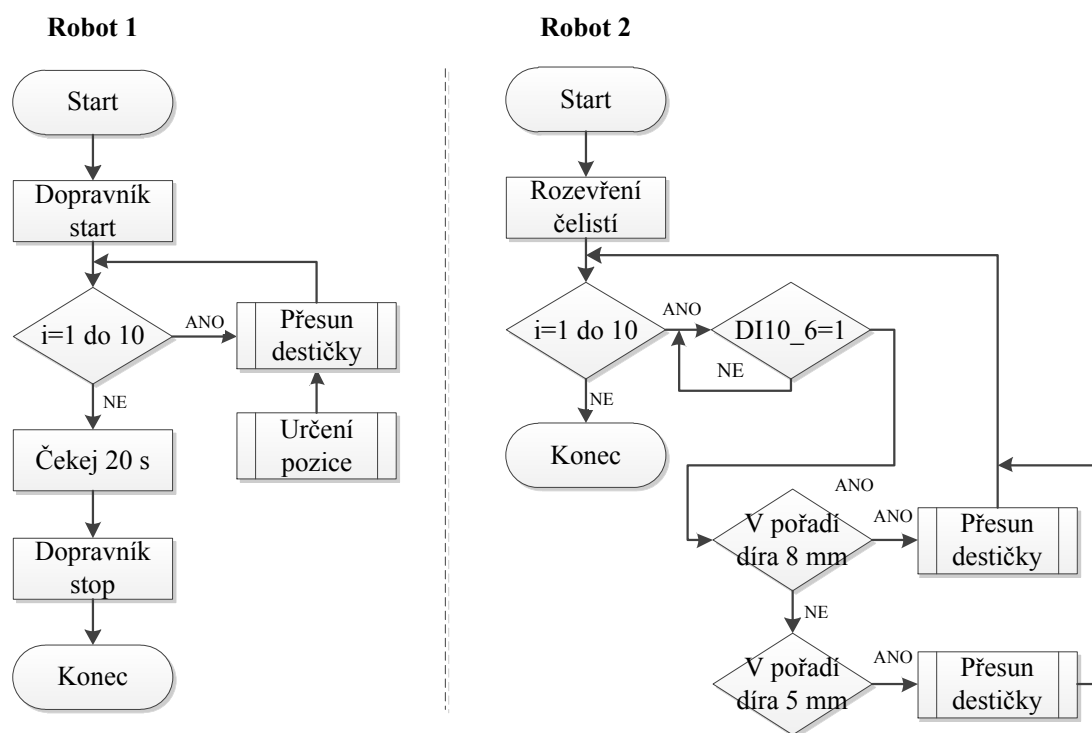
Obr. 33 Vytřízené destičky

### 3.5.1 Program pro roboty

**V úloze se bude řešit:**

- tvorba logiky v simulaci,
- použití funkce Smart Component,
- tvorba programu s ohledem na spolupráci se systémem strojového vidění
- upload programu do reálné řídicí jednotky.

Na Obr. 34 je zobrazen vývojový diagram pro základní chování programu při třídění destiček. První robot vykládá destičky z jednoho sloupce. Při každém dalším vyložení snižuje robot svoji polohu o výšku destičky. Po posledním vyložení následuje pauza 20 s pro dojezd poslední destičky k druhému robotu.



Obr. 34 Vývojový diagram pro třídění destiček

Druhý robot musí být schopen přijímat informaci vyslanou z rozšiřujícího modulu I/O systému strojového vidění. Může nastat situace, kdy robot bude zrovna vykonávat pohyb a kamerový systém vyšle přes digitální výstup signál s informací o typu destičky. Tento problém lze řešit pomocí přerušení (interrupt). Právě přerušení umožňují programu reagovat na událost nezávisle na instrukci, která se zrovna provádí. Pokud nastane událost, v tomto případě změna digitálního vstupu řídicí jednotky, program je přerušen a vykoná se

speciální TRAP rutina. Po jejím vykonání se pokračuje v programu od místa přerušení.  
[16]

```

...
!Prměnná intnum uchovává specifickou hodnotu, podle které se identifikuje
přerušení.
var intnum prerusenil;

...

TRAP trapSignal
!Tato rutina se vykoná, pokud nastane přerušení1.
Rutina;
RETURN;
ENDTRAP

...

PROC Main()

...

!Při inicializaci je třeba přiřadit přerušení TRAP rutině. Na jednu TRAP
rutinu je možné přiřadit několik přerušení.
CONNECT prerusenil WITH trapSignal;
!Pokud se změní signál DI10_10 spustí se přerušení a vykoná se napojená
TRAP rutina. V tomto případě signál vychází z I/O modulu CVS. Signál je
vyslán v případě, že je identifikována destička s dírou o průměru 8 mm.
ISignalDI DI10_10, 1, prerusenil;

...

!Příkaz IDelete zruší přerušení.
IDelete prerusenil;
ENDPROC

...

```

Možnosti přerušení jsem využil také v řídicí jednotce prvního robotu. Do robotu vede signál optické závory, který spouští přerušení. Následně je s časovou prodlevou vyslán signál do I/O modulu systému strojového vidění, který spouští zachycení obrazu.

```

...

!Rutina vystupSenzoru
TRAP vystupSenzoru
if DI10_3=1 then !DI10_3 - optická závora
!Prodleva pro vysokou rychlost dopravníku je zvolena tak, aby byl předmět
ve vhodné pozici pro zachycení snímku.
WaitTime 0.65;
!Výstup na kamerový systém spustí okamžité zachycení snímku.

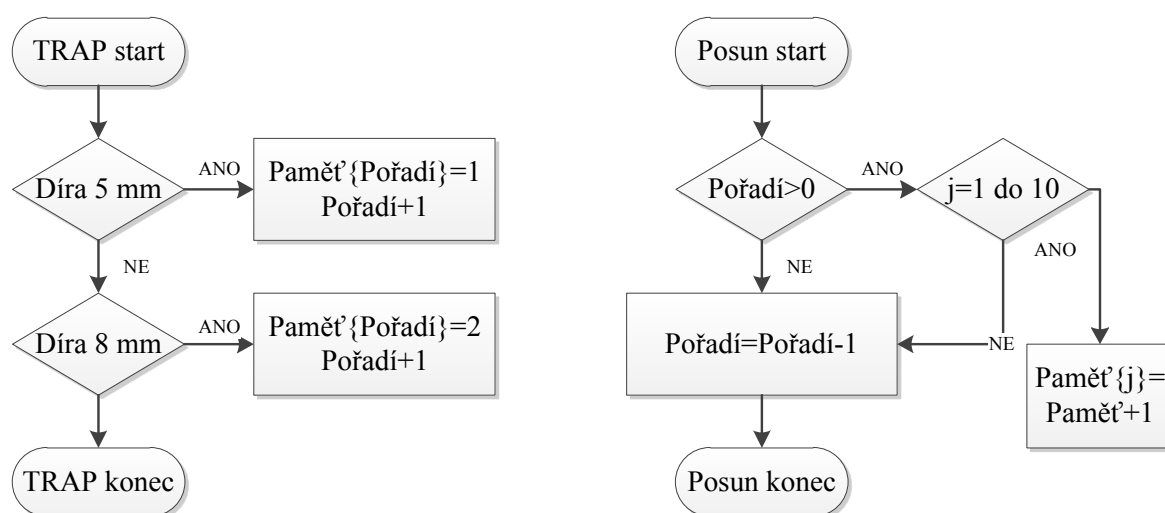
```

```
SetDO DO10_8, 1;
RETURN;
ENDTRAP
```

...

Robot, který odebírá destičky, dělí od kamerového systému tři metry. Při provozu dochází k hromadění několika destiček mezi oběma systémy. Bez vytvoření paměti, ve které je uloženo pořadí všech destiček, které byly již identifikovány a jsou na cestě k odebrání, by nebylo možné předměty třídit.

### Robot 2



Obr. 35 Diagram pro TRAP a Posun paměti po uložení destičky

Paměť byla vytvořena jako pole typu num. Po spuštění TRAP rutiny se vyhodnocuje podmínka. Pokud se jedná o signál, který představuje informaci o destičce s dírou 5 mm, zapíše se do paměti na první místo pole hodnota 1. Pro destičku s dírou 8 mm se zapíše hodnota 2. Po zápisu se zvýší proměnná pořadí, aby bylo možné při dalším zápisu ukládat na neobsazenou pozici v paměti.

```
!Paměť při inicializaci pro 12 pozic:
pamet:= [0,0,0,0,0,0,0,0,0,0,0,0,0,0]
!Paměť při zapsání tří destiček v pořadí [5 mm, 8 mm, 8 mm]:
pamet:= [1,2,2,0,0,0,0,0,0,0,0,0,0,0]
```

Po projití destičky optickou bránou robot ukládá uchopený objekt na jednu ze dvou pozic podle čísla v paměti. Po uložení destičky již není třeba informací o právě uloženém objektu uchovávat a je spuštěna rutina „Posun“ pro vymazání této informace.

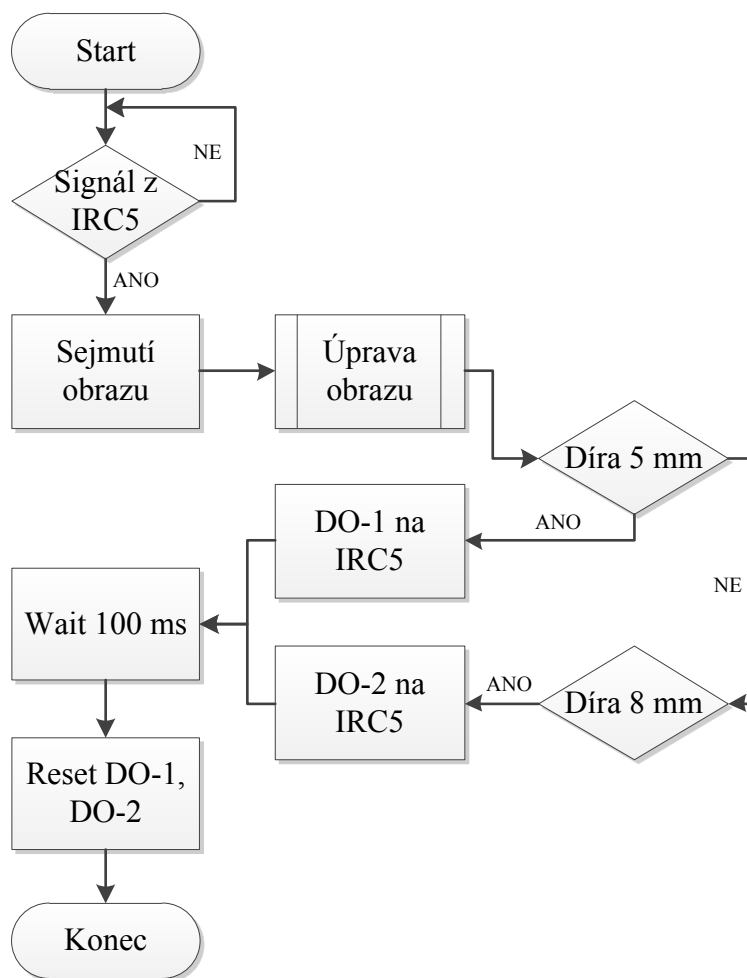
```
!Výchozí stav paměti:
pamet:= [1,2,2,0,0,0,0,0,0,0,0,0]
!Úprava podle vzorce pamet{j}:= pamet{j+1} pro první pozici:
pamet{1}:= pamet{2}
!Po proběhnutí kompletního cyklu:
pamet:= [2,2,0,0,0,0,0,0,0,0,0,0]
!Aktuální pořadí, do kterého se bude zapisovat je v tuto chvíli
„pamet{4}:=x“! Z toho důvodu se provede úprava pořadí:
poradi:=poradi-1;
```

### 3.5.2 Program pro CVS

Na Obr. 36 je zobrazen vývojový diagram pro systém strojového vidění pro úlohu třízení destiček podle průměru vyvrtané díry. Při vstupu OM do optické brány je okamžitě vyslán signál do řídicí jednotky robotu. V programu robotu po přijetí signálu nastává prodleva 0,65 s a poté je vyslán signál do systému strojového vidění a nastává sejmutí snímku. Prodleva zajišťuje čas nutný pro posun OM po dopravníku do polohy vhodné pro pořízení snímku. Hodnota prodlevy 0,65 s je dimenzována pro nejvyšší rychlost dopravníku.

#### V úloze se bude řešit:

- připojení k systému strojového vidění (CVS),
- sejmutí a příprava obrazu,
- komunikace pomocí přídatného modulu I/O.

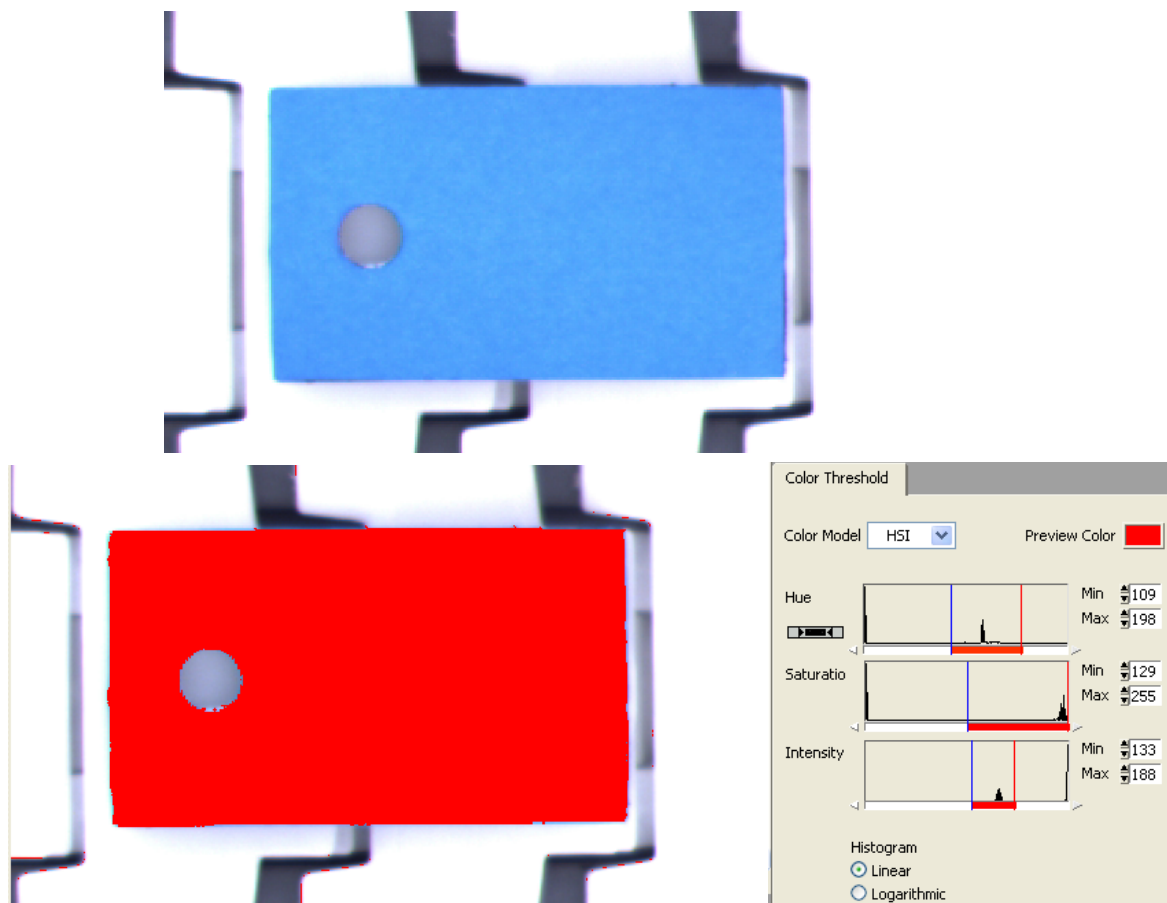


Obr. 36 Vývojový diagram pro systém strojového vidění

V programu National Instruments Vision Builder for Automated Inspection jsem zdrojový obraz nejprve upravil pomocí funkce „Color Treshold“, která umožňuje filtrovat barvy podle barevných prostorů RGB, HSL, HSV, HSI. Modely HSL, HSV, HSI jsou často používány pro strojové vidění a analýzu obrazu. [17]

- RGB (Red, Green, Blue) – červená, zelená modrá,
- HSV (Hue, Saturation, Value) - tón barvy, sytost, jas, [18]
- HSL (Hue, Saturation, Lightness) - tón barvy, sytost, světlost, [18]
- HSI (Hue, Saturation, Intensity) - tón barvy, sytost, intenzita. [19]

V mém případě jsem zvolil metodu HSI, která se ukázala jako optimální pro výběr destičky a odfiltrování tmavého prostoru mezi články dopravníku. (Obr. 37)

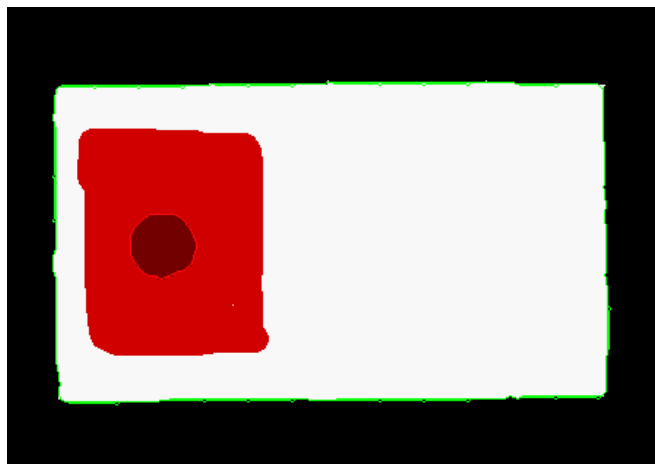


Obr. 37 Zdrojový obraz z CVS (nahore), filtrování pomocí modelu HSI (dole)

Vzniklý obraz jsem vyčistil funkcí „Remove small objects“ od malých segmentů na krajích článků dopravníku.

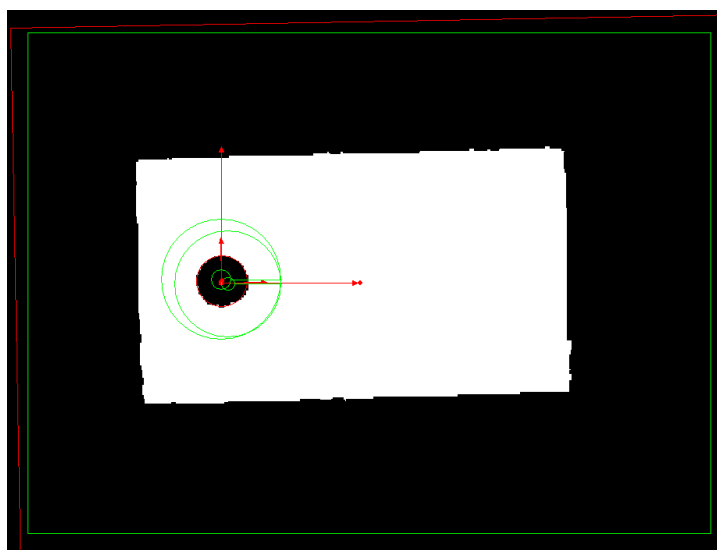
Pro rozpoznání destičky jsem použil funkce „Geometric Matching“. Tato funkce umožňuje hledat objekt na základě jeho geometrických vlastností. V obraze je možné definovat oblasti, které se budou ignorovat. Na Obr. 38 lze vidět identifikované křivky zeleně a ignorovanou oblast červeně.





*Obr. 38 Hledání geometrického vzoru*

Rozpoznání průměru díry je realizováno pomocí funkce pro hledání kružnice. Funkcí je možné měřit průměr nalezených kružnic. V defaultním nastavení je vzdálenost měřena v pixelech. Je možné kalibrovat CVS tak, aby měřené jednotky byly v metrickém systému. Cílem inspekce je rozlišovat dva průměry. Pro tento úkon bylo dostačující rozlišování v pixelech.



*Obr. 39 Hledání kružnic*

Funkce hledání kružnice je použita dvakrát (Obr. 39). Díra s průměrem 5 mm se jeví jako 28 pixelů, díra s průměrem 8 mm jako 45 pixelů. Při prvním hledání kružnice je maximální hledaný průměr omezen na 35 pixelů. Pokud je hledání neúspěšné, výsledek kroku nabývá hodnoty false. Tento princip je využit při druhém hledání kružnice, kdy je minimální průměr nastaven na 40 pixelů. Informace shrnuje Tab. 11.

Průměr díry	Průměr [pix]	Hledání kružnice 1	Hledání kružnice 2
5 mm	28	TRUE	FALSE
8 mm	45	FALSE	TRUE

*Tab. 11 Výstup z hledání kružnic*

K předání informace o průměru byl použit modul I/O systému strojového vidění. Pokud je hledání jedné z děr úspěšné (stav true), vyšle se z modulu digitální výstup na řídicí jednotku robotu (DO1 pro Ø 5 mm, DO2 pro Ø 8 mm). Zpracování informace v robotu je popsáno v kapitole 3.5.1.

### 3.6 Úloha třízení dominových kamenů

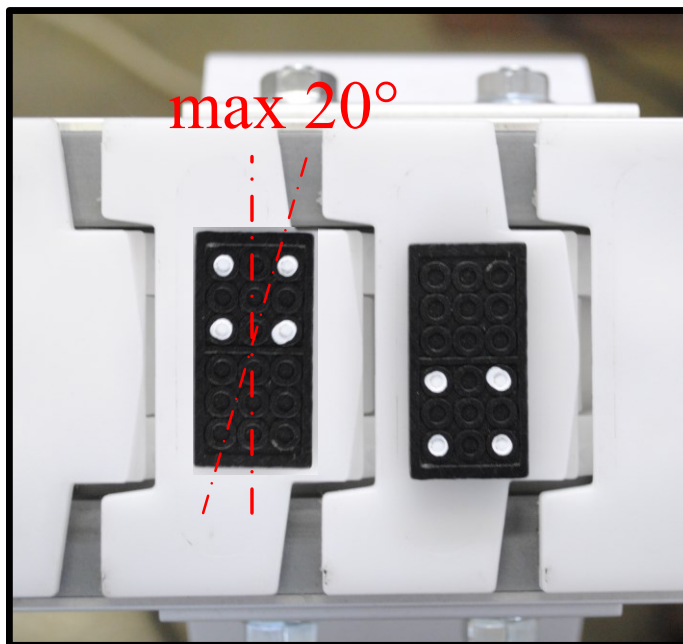
V úloze je objektem manipulace kompletní dominový balíček s 28 kameny. Hlavní myšlenkou úlohy je rozpoznávání počtu teček na kamenech, které přicházejí po dopravníku. Kamene jsou posléze vykládány robotem na odkládací stůl do definované matice (Obr. 40). Kamene jsou na dopravník vykládány ručně z důvodu nemožnosti uchopení stávajícím přísavkovým efektozem.



*Obr. 40 Vyskládané kameny*

Úloha využívá pokročilých funkcí systému strojového vidění. V úloze třídění destiček byla informace přenášena přes NI Vision I/O Terminal Block. Pomocí maximálního počtu DO (4x) je možné přenést informaci o 16 kamenech ( $2^4$ ). Pro přenos informací byl z důvodu tohoto omezení zvolen sériový port.

V úloze bylo řešeno natočení kamenu o  $180^\circ$  s tolerancí  $20^\circ$  od ideální polohy (Obr. 41). V programu je také nutné ošetřit chybné inspekce a neidentifikované kameny odebírat.



*Obr. 41 Možné umístování kamenů*

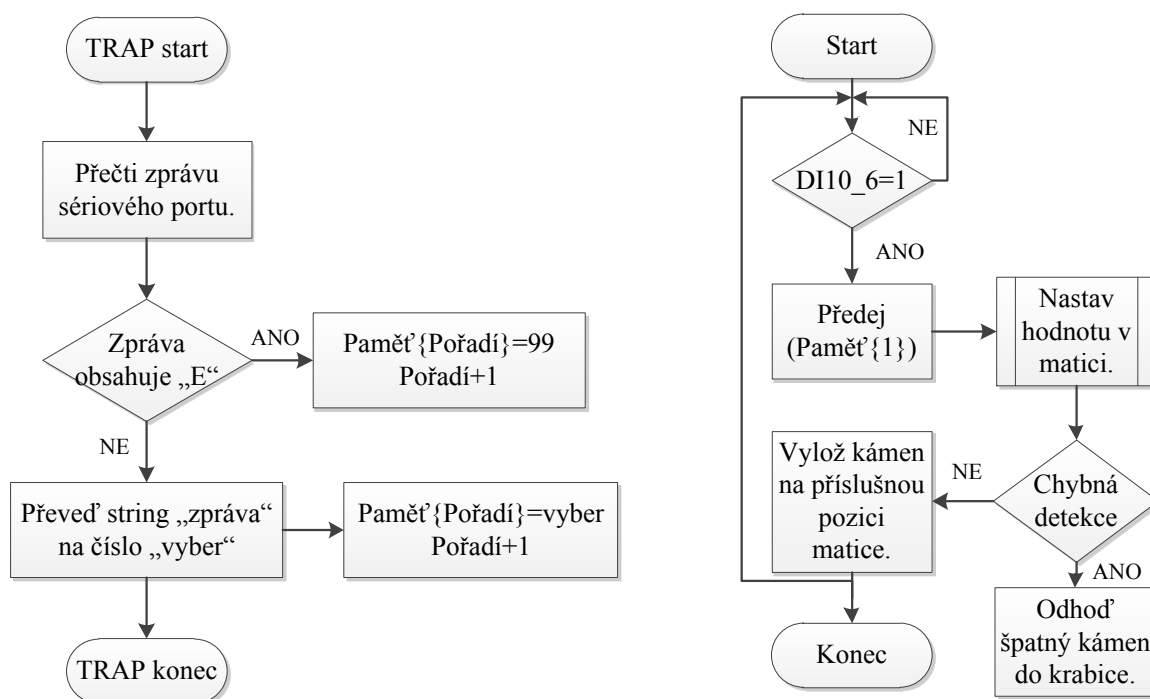
### 3.6.1 Program pro roboty

Algoritmus programu pro robot, který odebírá kameny a ukládá je do matice, byl navrhnut na základě předchozích znalostí programovacího jazyka RAPID. Funkčnost algoritmu pro ukládání kamenů byla odzkoušena v RobotStudios. K této úloze nebyla vytvořena podrobná simulace, ve které by například nešlo simulovat informace posílané sériovým portem, a tím pádem by se simulace od reálného programu značně lišila a sloužila spíše k prezentačním účelům.

Program pro robot, který tvoří z kamenů matici, byl rozdělen do tří modulů:

- Kalibrace.mod – definice efektoru, souřadnice workobjectů,
- KostkyDatabase.mod – převod a příprava informace pro Vylozeni.mod,
- Vylozeni.mod – hlavní modul.

Na Obr. 42 je zobrazen zjednodušený vývojový diagram pro chování programu při třídění dominových kamenů. Rutina TRAP slouží ke čtení zprávy ze sériového portu a ukládání informace do paměti. Paměť funguje na stejném principu jako v úloze třídění destiček. Pro úlohu bylo nutné ošetřit chybné inspekce. Ty je nutné do paměti zapisovat také a chybně identifikované dominové kameny jsou odebírány a odhazovány do krabice s chybnými kusy.



Obr. 42 Zjednodušený diagram pro ukládání kamenů do matice

V modulu „KostkyDatabase.mod“ je vytvořena matice{7,7}. V modulu jsou nastaveny akce pro 57 zpráv, které mohou přijít ze sériového portu. 2x 28 pro správně detekované kameny a jedna zpráva pro chybové stavy. Zpráva má vždy délku dva znaky. Správné inspekce jsou v číselném formátu, kde první číslo představuje počet teček na jedné polovině kamenu, druhé číslo zase na druhé polovině. Tímto uspořádáním čísel se rozlišují kameny, které jsou otočeny o 180° od ideální polohy. Pokud dojde k chybovému stavu při inspekci, ve zprávě se odešle znak „E“ značící chybu.

```

!Ukázka z modulu KostkyDatabase.mod
!Příslušné místo v matici se přepíše na hodnotu 1 nebo 2 dle natočení
PROC VyberKostky(num kostky) !Do argumentu „num kostky“ se dosadí zpráva
...
case 56:
matice{2,6}:=1; !kámen bez natočení

```

```

case 65:
matice{2,6}:=2; !kámen natočený o 180°
case 66:
matice{1,7}:=1;
!Ošetření chybových stavů 99 značí chybu, matice je datového typu „num“ a
nebylo by možné do ní možné uložit string „E“
case 99:
matice{7,7}:=1;

...

ENDPROC

...
```

Hlavní část ukládání kamenů probíhá v modulu „Vylozeni.mod“. Při spuštění programu proběhne rutina „Inicializace“, ve které se vytvoří matice robtargetů nazvaná „targetArray“. Samotné ukládání vykonává rutina „Uloz“.

```

PROC Uloz()
!Aktualizuje matici v modulu KostkyDatabase.mod na první kámen v pořadí.
VyberKostky(pamet{1});

...

!Cyklus projede všechny pozice v matici (z modulu KostkyDatabase.mod).
Pokud je v matice{i,j} rovna 1 nebo 2, proběhne ukládání kamenu.
for i from 1 to xRadky do
  for j from 1 to yRadky do
    if matice{i,j}=1 then
      !Kámen je bez natočení.
      targetArray{i,j}.rot:=OrientZYX(0,0,0);

      ...

      !Pohyb na žádané místo.
      MoveL targetArray{i,j},v50,fine,PGN80Plu_wobj\WObj:=ukladani;
    elseif matice{i,j}=2 then
      !Kámen je otočený o 180°. Natočí se souřadný systém efektoru.
      targetArray{i,j}.rot:=OrientZYX(180,0,0);

      ...

    endif
  endfor
endfor

...

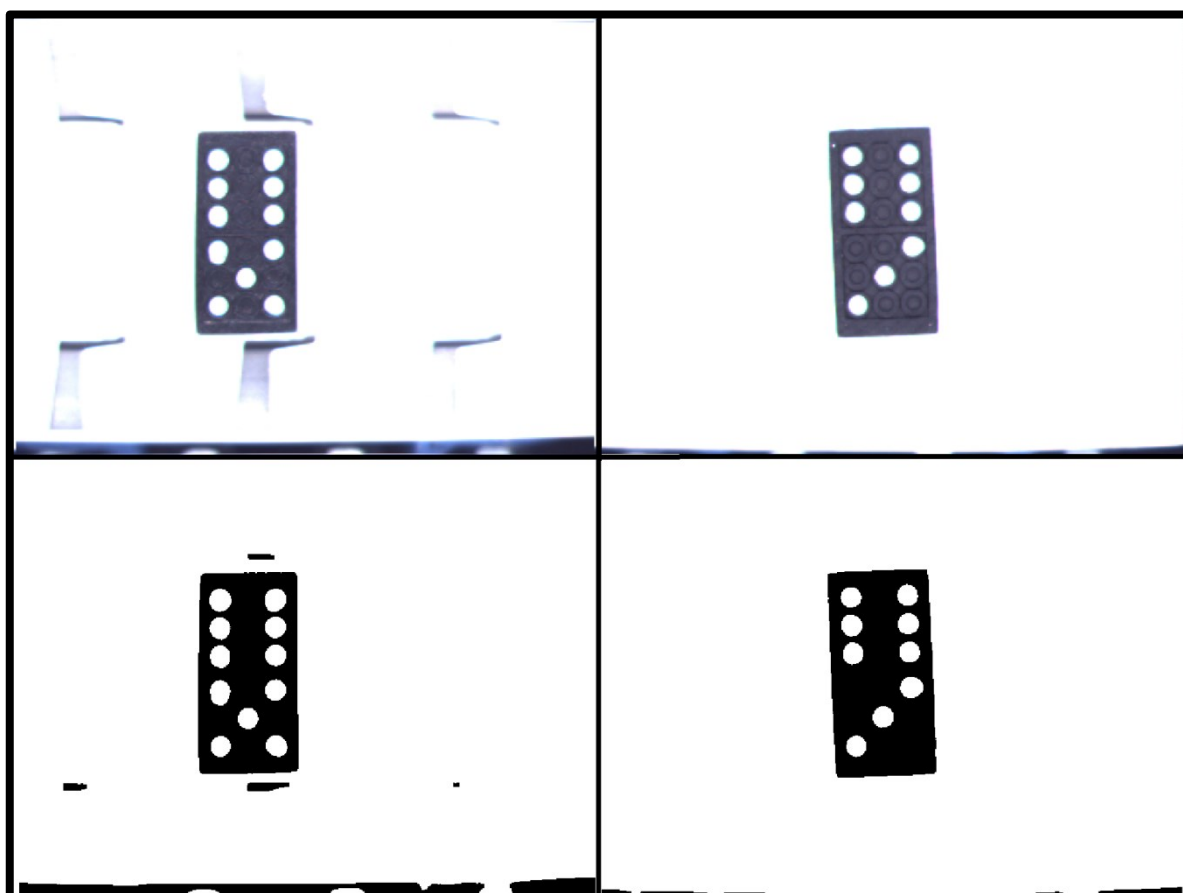
ENDPROC
```

Rutina „Ulož“ je univerzální pro všechny kombinace kamenů a použití matice robtargetů značně snižuje počet řádků kódu. Detailní komentovaný kód k programu obsahuje příloha D.

Z důvodu spouštění zachycení snímku z kamery pomocí signálu z robotu s přísavkovým efektozem je nutné vytvořit program i pro tento robot, i když nedochází k pohybům s robotem. Problematika zachycení snímku byla blíže popsána v kapitole 3.5.1. Dále je nutné tímto robotem ovládat dopravník. Dopravník je automaticky zastaven, pokud přes optickou bránu u CVS neprojde během dvaceti sekund objekt manipulace. Při každém průchodu objektu bránou se uběhnutý čas resetuje. Funkce pro stopování času byla použita a je blíže popsána v kapitole 3.3. Detailní kód obsahuje příloha D.

### 3.6.2 Program pro CVS

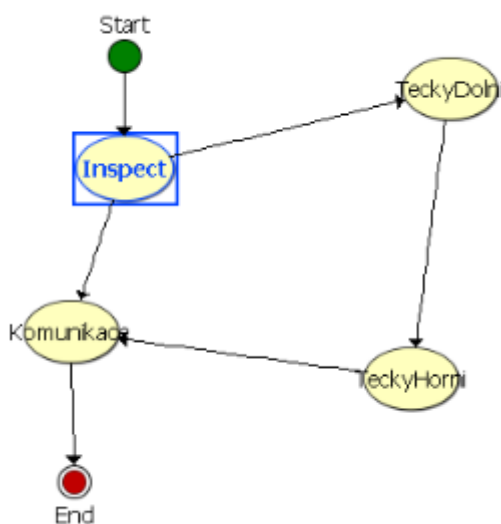
Při získávání obrazu pro inspekci nastal problém s kovovým rámem, který je pod články dopravníku viditelný. V sejmutém obrazu vznikaly segmenty barevně velmi podobné černým kamenům (Obr. 43 vlevo).



*Obr. 43 Zdrojový (nahore) a filtrovaný (dole) obraz dominových kamenů*

Pomocí dostupných filtrů nebylo možné dosáhnout odstranění těchto segmentů. Při některých inspekcích tyto segmenty splynuly s kameny a docházelo k nerozpoznání kamenu. Pro odstranění tohoto problému byla mezi segmenty a rám nalepena bílá páska, která problém segmentů odstranila (Obr. 43 vpravo).

Program je rozdělen na čtyři kroky (Obr. 44). V kroku „Inspect“ probíhá sejmutí, úprava obrazu a měření intenzity. Pokud dojde k chybě při identifikaci, pokračuje program do části „Komunikace“ a odešle na sériový port zprávu „EE“ značící chybný stav. Při zdařilé identifikaci pokračuje program z kroku „Inspect“ na krok „TeckyDolni“ a „TeckyHorni“. V těchto krocích probíhá logické rozhodování a odeslání zprávy o počtu teček. Blok „Komunikace“ vyšle digitální signál do řídicí jednotky robotu. Na základě tohoto signálu se v robotu spouští čtení zprávy ze sériového portu.



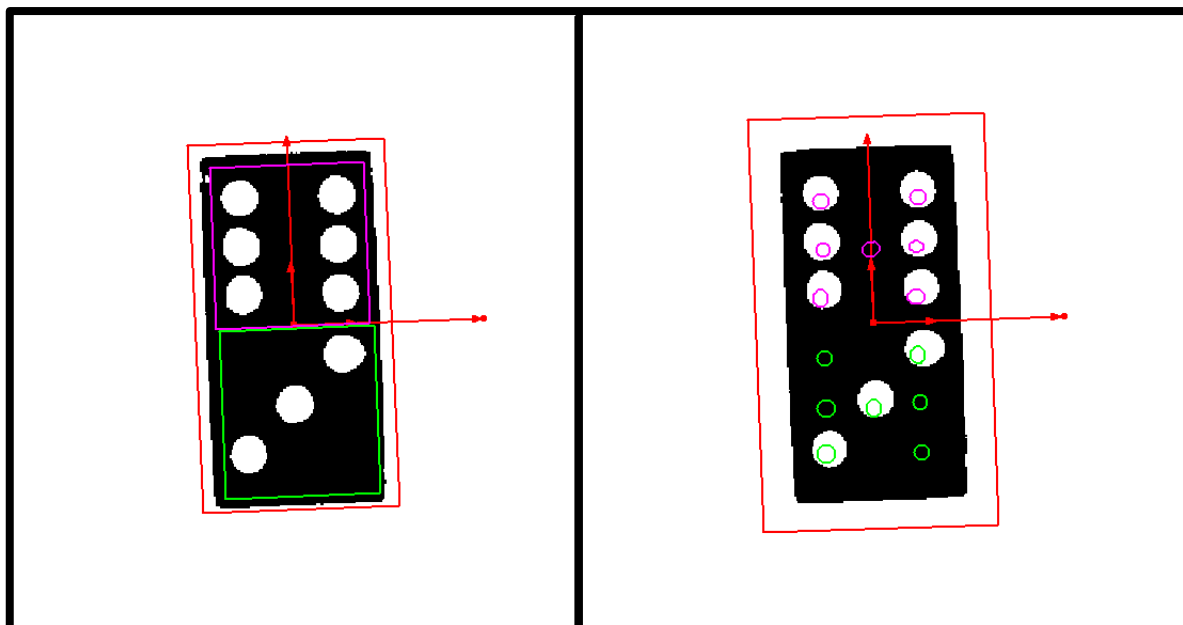
Obr. 44 Rozdělení programu

Dominové kameny jsou identifikovány pomocí metody geometrického vzoru, která byla použita u třízení destiček s dírou.

Pro rozpoznání počtu teček byla použita funkce měření intenzity barev v černobílém obrázku. Funkce vrací hodnotu 0 - 255 (černá - bílá) v měřené oblasti. Měření intenzity ve čtvercových polích se ukázalo jako vysoce nespolehlivé (Obr. 45 vlevo). Při prvním průzkumu tato metoda vypadala nadějně, protože hodnoty intenzity pro různý počet teček se nacházely sice v těsných intervalech, ale rozpoznávání na několika kusech proběhlo podle očekávání. Při ostrém chodu programu však docházelo ke spoustě vadných inspekcí



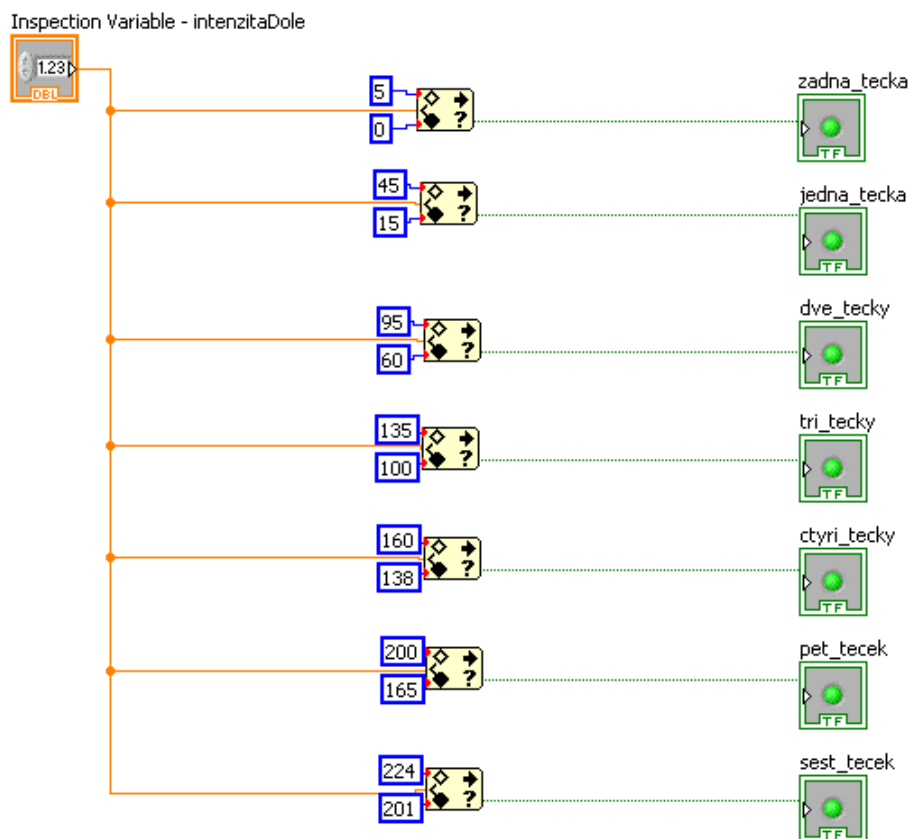
a při bližším průzkumu jsem zjistil, že rozestup intervalu je nedostatečný. Jako optimální se ukázalo měření intenzity v kruhových polích na místě předpokládaných teček (Obr. 45 vpravo). Při tomto způsobu je rozpětí intervalu mezi hodnotami dostatečně velké.



*Obr. 45 Měření intenzity barev*

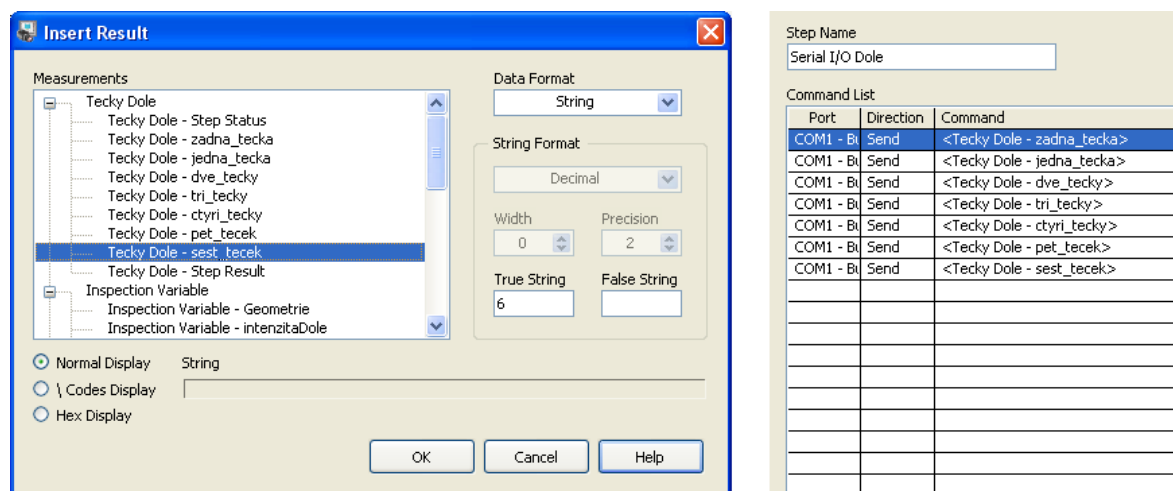
Pro vyhodnocení intenzity jsem použil funkci „Calculator“, která umožňuje použít početních i logických operací. Pro nezkušeného uživatele je toto prostředí neintuitivní a grafická podoba prostředí práci příliš neusnadňuje. Na Obr. 46 je vytvořena logická vrstva pro rozhodování, kolik teček je v měřené oblasti. Vstupním parametrem je naměřená intenzita. U ní se zjišťuje, zda se nachází v jednom ze šesti intervalů. Pokud ano, výsledkem je hodnota „true“ u příslušného logického výstupu (Obr. 46 vpravo).

Logické rozhodování probíhá zvlášť pro horní a dolní část dominového kamenu (z pohledu kamery). Rozdělení intervalů je stejné, liší se pouze vstupní a výstupní hodnoty proměnných.



Obr. 46 Logická vrstva při rozhodování o počtu teček

Při odesílání informace přes sériový port se využívá logického stavu příslušné proměnné s počtem teček. Na Obr. 47 vlevo lze vidět nastavení pro zprávu pro šest teček. Pokud proměnná nabývá hodnoty „true“ odešle se zpráva obsahující číslici „6“. V opačném případě je zpráva prázdná a nic se neodesílá.

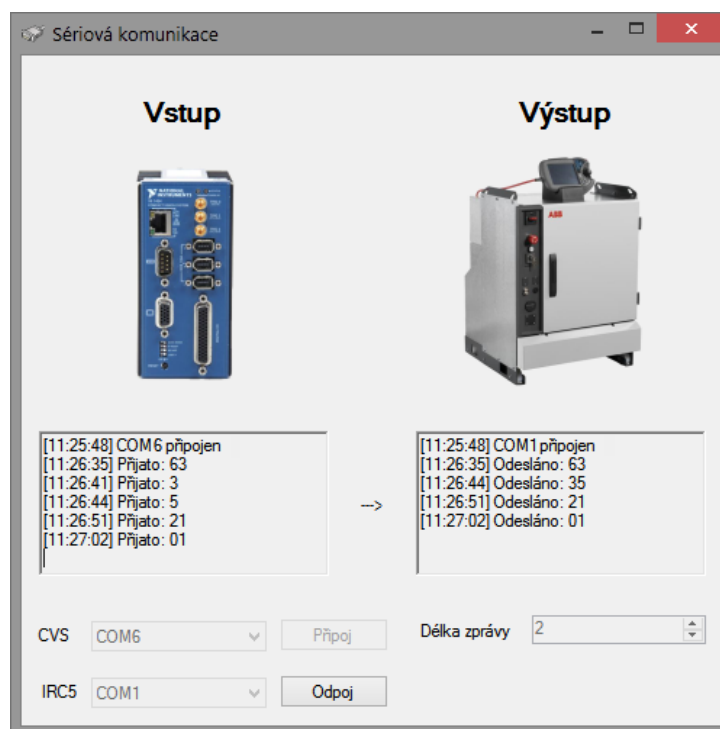


*Obr. 47 Sériová komunikace*

Takto nastavená logika je i u ostatních stavů (Obr. 47 vpravo). Sériová komunikace probíhá zvlášť pro horní a dolní pole teček. Pro každé pole se tedy odešle zpráva o délce jednoho znaku.

### 3.6.3 Program pro komunikaci přes sériový port

V rané fázi diplomové práce při zjišťování stavu pracoviště jsem se začal mimo jiné učit programovací jazyk C#. Nabyté znalosti o programovacích principech mi pomohly při tvorbě RAPID kódu pro roboty. Rozhodl jsem se vytvořit program pro přeposílání komunikace po sériovém portu pro potřeby úlohy třídění dominových kamenů (Obr. 48). Program jsem se snažil vytvořit tak, aby se dal případně využít univerzálně pro další úlohy vyžadující posílání zpráv z CVS na IRC5 přes sériový port.



Obr. 48 Program pro přeposílání zpráv

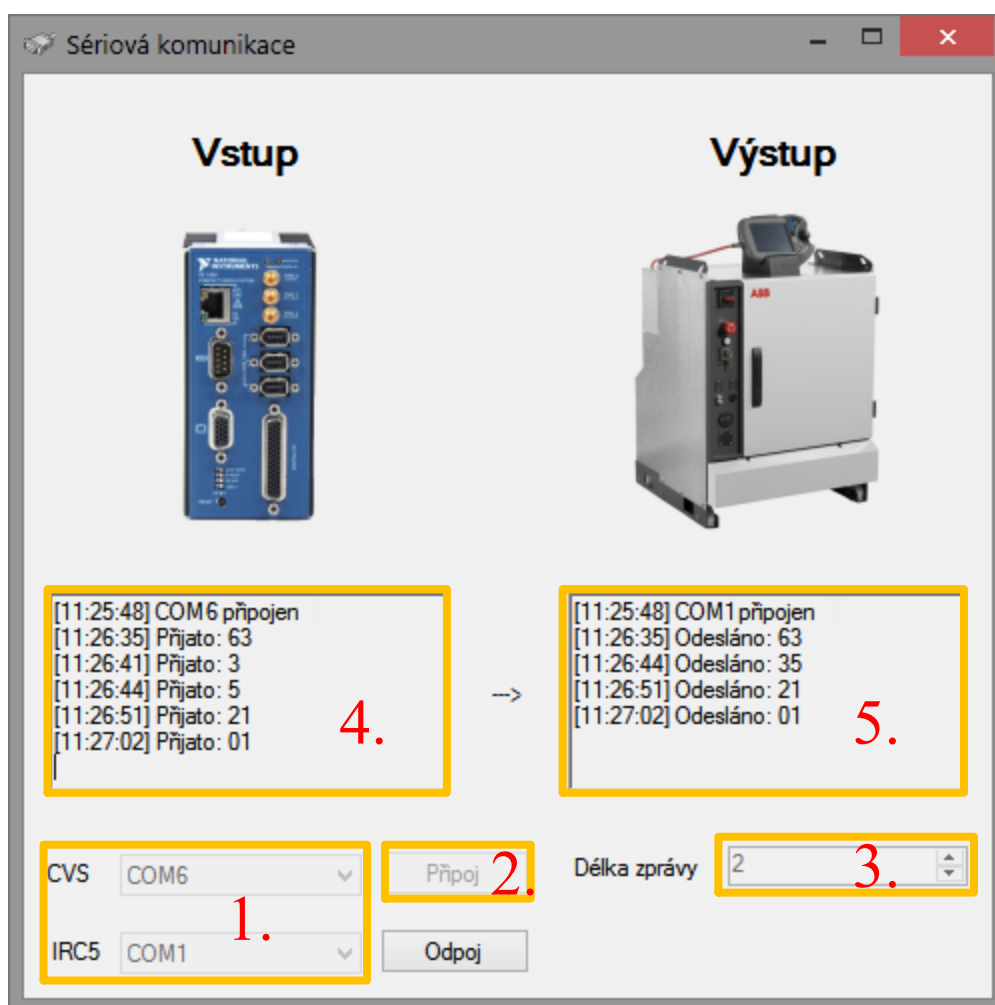
V programu byly ošetřeny chybné uživatelské vstupy:

- výběr stejného portu pro CVS a IRC5,
- připojení k již otevřenému portu,
- prázdný výběr v poli CVS nebo IRC5.

Zjistil jsem, že zpráva z CVS délky dva znaky se při zachycení v mém programu někdy dělí na části. Abych dosáhl pokaždé stejné délky zprávy odesílané na IRC5, přidal jsem do programu možnost volby délky zprávy. Pokud přijde zpráva z CVS na části (Obr. 49 (4.)), program ji neodešle dřív, než dojdou všechny potřebné znaky (Obr. 49 (5.)). Detailní popis zdrojového kódu je v příloze D.

Hlavní prvky programu (Obr. 49):

1. rolovací nabídka s dostupnými porty,
2. tlačítko pro připojení k portům,
3. okno pro nastavení délky zprávy,
4. textové okno s výpisem přijatých dat,
5. textové okno s výpisem odeslaných dat.



Obr. 49 Hlavní prvky programu

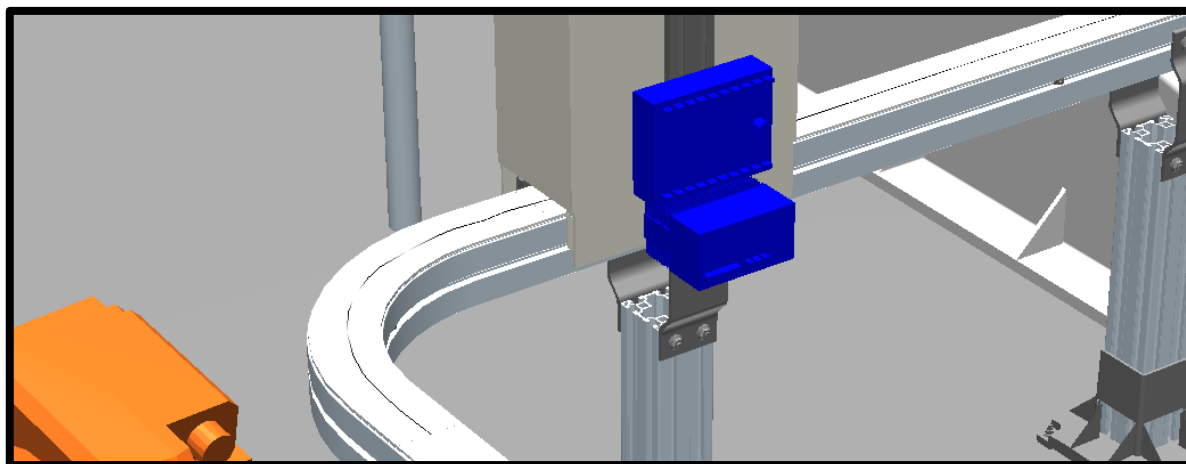
## 4 Smart Component

Při zkoumání dříve zpracovaných simulací v RobotStudiu na katedře, jsem zjistil, že pro simulaci různých událostí jako například pohyb s objektem se využívá Event Manager. V něm se nastaví akce a k ní se přiřadí virtuální digitální signál, který ji spustí. Velkou nevýhodou je, že pro každý objekt, se kterým chceme provádět akci, musíme mít vlastní řídicí signál.

RobotStudio od verze 5.14 obsahuje nástroj Smart Component, který má sloužit jako náhrada za Event Manager. Smart Componentům lze nastavovat samostatnou vnitřní logiku, senzory přítomnosti objektu, přichycení a odpojení objektu atd. Velkou výhodou je možnost přenášet například efektory do jiné stanice bez ztráty funkcí pro přichycení a odchycení jakékoliv uchopované součásti. Smart Component typu přísavka a zdroj OM využil Ing. Jakub Mžík v diplomové práci. [6]

### 4.1 SC dopravník

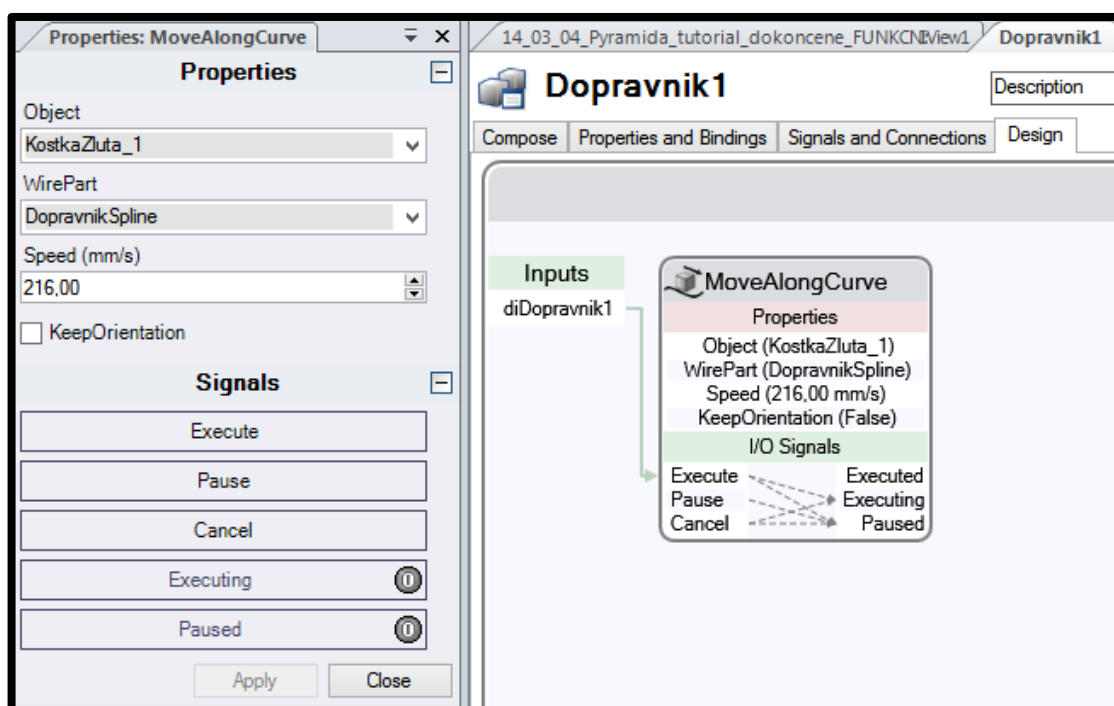
Při tvorbě simulace jsem narazil na problém při vytváření dopravníku se zatáčkami. Nejprve jsem se pokoušel nastavit logiku dopravníku v Event Manageru. Po množství neúspěšných pokusů jsem hledal cestu, která by umožňovala pohybovat s předmětem po libovolné trajektorii. Po prozkoumání vlastností Smart Component jsem vybral vhodnou funkci *MoveAlongCurve*, která umožňuje pohybovat předmětem nastavenou rychlostí po zvolené křivce.



Obr. 50 Křivka na dopravníku

Ke Smart Componentě jsem přidal ještě senzor přítomnosti OM, který jsem umístil na pozici vykládání kostek na dopravníku. Hlavní myšlenkou bylo používat každý detekovaný předmět po definované trajektorii. První detekovaná kostka se chovala podle předpokladu. Druhá kostka se po detekci sice začala hýbat, ale její pohyb se spojil s pohybem první a řešení se ukázalo jako nerealizovatelné.

Jako optimální řešení se nakonec ukázalo vytvoření samostatné Smart Componenty typu dopravník pro každou kostku zvlášť.



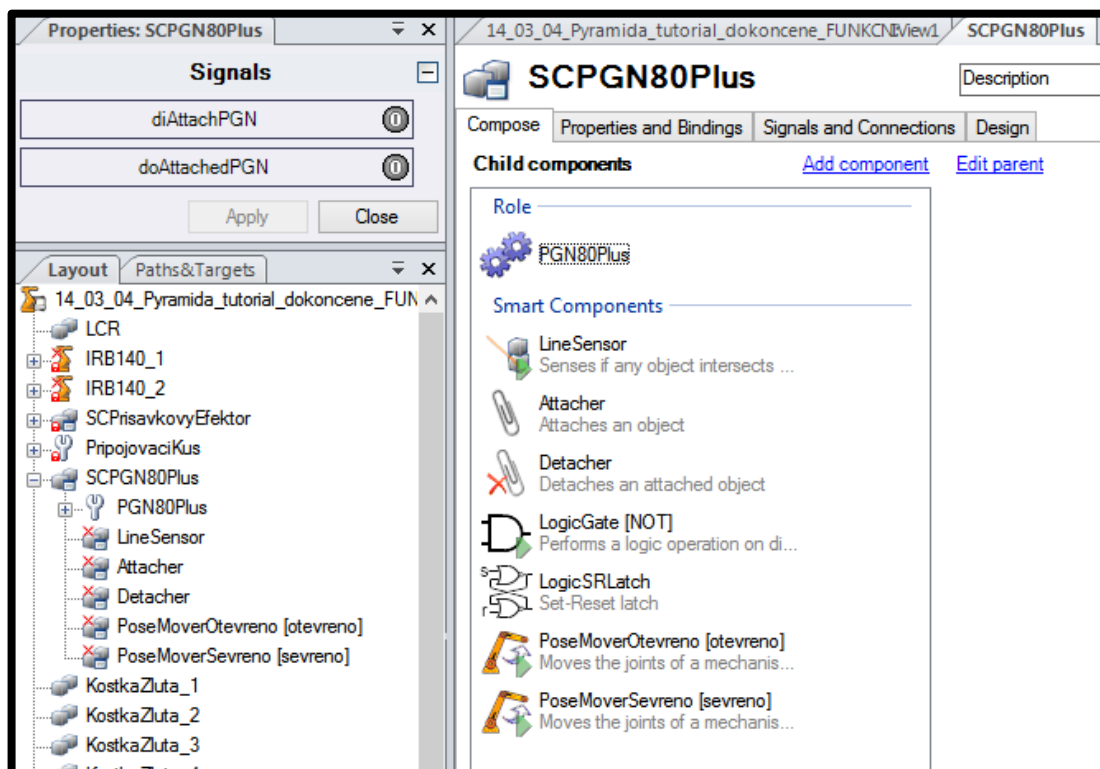
Obr. 51 Smart Componenta simulující dopravník

## 4.2 SC efektor PGN80 plus

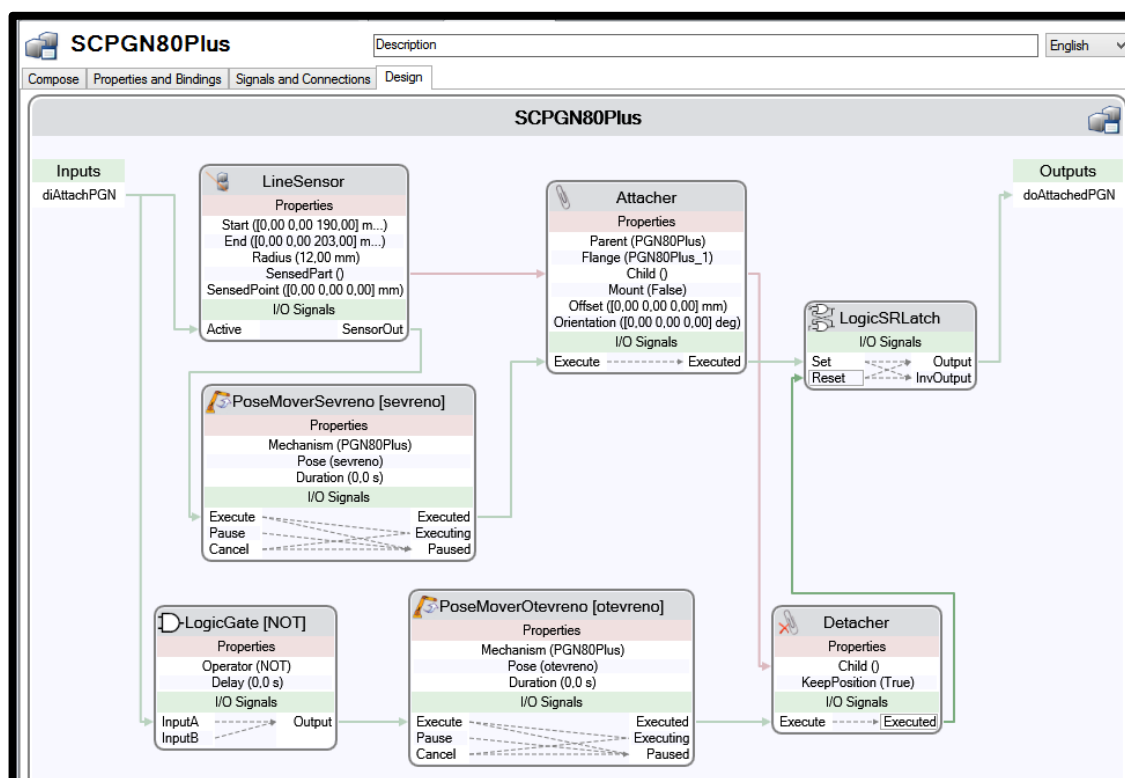
Čelistový efektor přichytí detekovaný OM a přesune čelisti do polohy sevřeno. Po umístění OM odepne a otevře čelisti. Skládá se ze sedmi funkcí:

- LineSensor – slouží k detekci OM,
- Attacher – přichytí detekovanou součást k efektoru,
- Dettacher – odepne OM,
- LogicSRLatch – slouží ke zpracování logického signálu, zda je OM přichycen,
- LogicGate [NOT] – vytváří log 1 pro odepnutí,
- PoseMover [sevřeno] – sevře čelisti,

- PoseMover [otvreno] – otevře čelisti.



*Obr. 52 Smart Component PGN80 Plus*

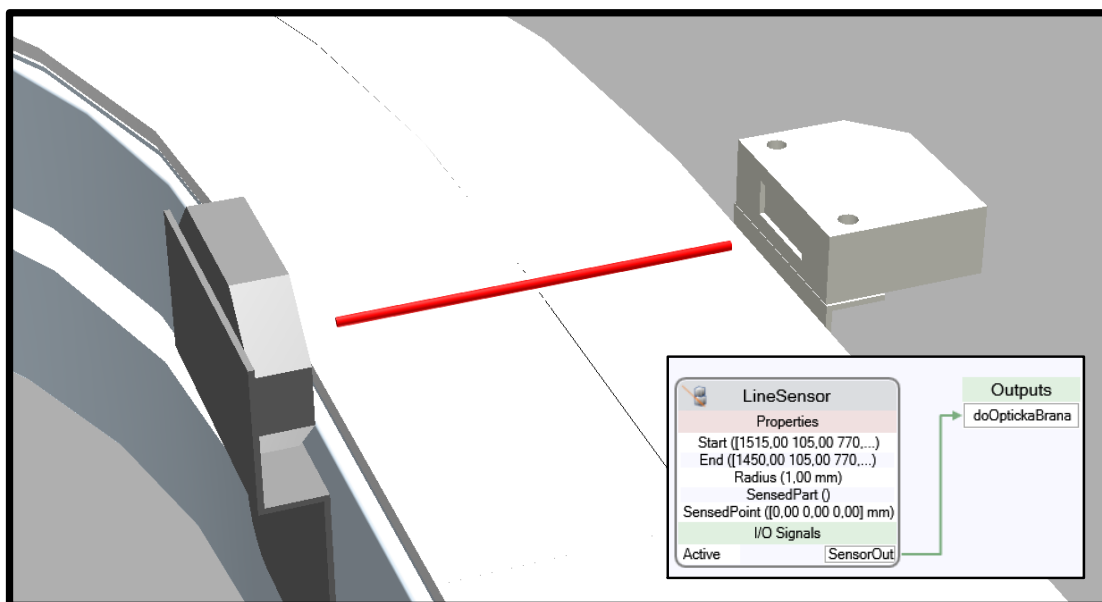


*Obr. 53 PGN80 Plus propojení logiky*



### 4.3 SC optická závora

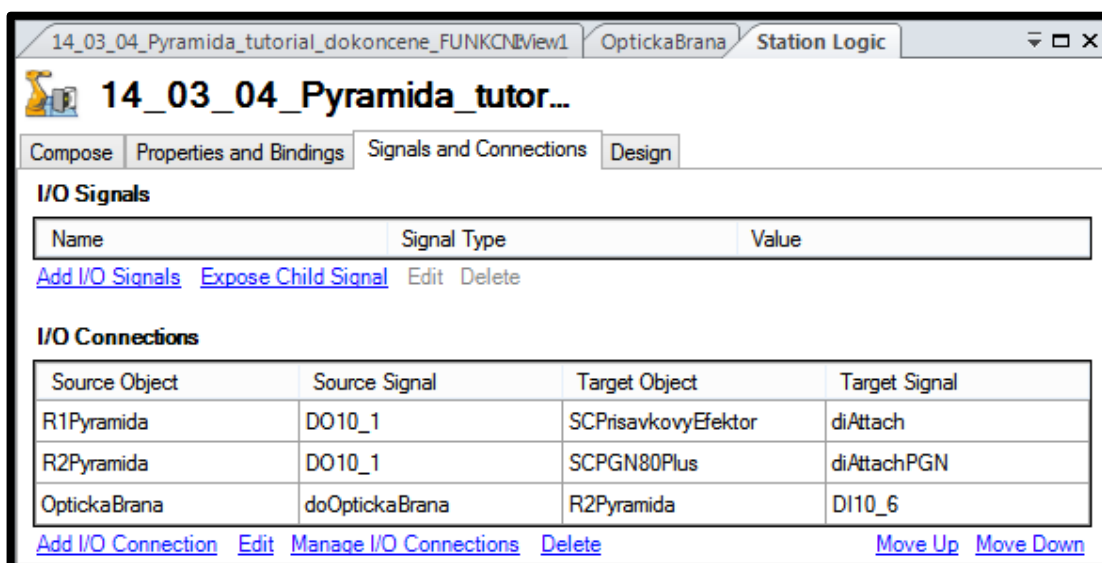
Pro simulaci optické závory jsem použil jediný prvek – LineSensor, který pouze detekuje průchod OM. Takto vytvořená závora věrně kopíruje chování reálné závory.



Obr. 54 Smart Component optická závora

### 4.4 Propojení signálů SC

Signály ze Smart Component je třeba přiřadit signálům z virtuální řídicí jednotky. (Obr. 55). Bližší informace v příloze B - návod pro simulaci – stavba pyramidy.



Obr. 55 Propojení signálů

## 5 Závěr

Diplomová práce se zabývá tvorbou výukových úloh pro pracoviště s roboty ABB IRB 140. V úvodu je popsán stav stávajícího robotizovaného pracoviště na Centru robotiky Katedry robototechniky VŠB-TU Ostrava. Pro jednotlivé komponenty byly zjištěny technické parametry a způsob jejich zapojení a ovládání.

Na základě zjištěných informací byly sestaveny požadavky. Byly vybrány vhodné objekty manipulace – dřevěné kostky, destičky s dírou a dominové kameny. Destičky s dírou byly vyrobeny svépomocí z PVC desky. Dominové kameny byly zakoupeny ve dvou kompletních baleních.

Hlavní část práce obsahuje popis řešených úloh. Dále obsahuje popis měření, které bylo využito pro stanovení přesné rychlosti dopravníku. V úvodu kapitoly je popis kroků vedoucích k zapnutí robotů a periférií na pracovišti.

První úloha – stavba pyramidy – je rozdělena na dvě části. První část je zaměřena na programování prostřednictvím dotykového prostředí – FlexPendantu. V druhé části je řešena tvorba simulace úlohy v RobotStudiu. V úloze je využito dvou robotů a článkového dopravníku. První robot vykládá na dopravník kostky ze zásobníku, druhý kostky za chodu odebírá a staví je do tvaru pyramidy.

Druhá úloha – třídění destiček s kruhovou dírou – využívá dva roboty, článkový dopravník a systém strojového vidění. K úloze je vytvořena simulace v RobotStudiu a program pro systém strojového vidění. Úloha demonstruje možnost rozpoznání průměru díry a třídění. Destičky jsou vykládány prvním robotem na dopravník a druhým jsou za chodu odebírány a třízeny do dvou sloupců.

Třetí úloha – třízení dominových kamenů – využívá pokročilé možnosti RAPID kódu a systému strojového vidění. V úloze jsou popsány problémy při analýze obrazu. Dominové kameny jsou odebírány z jedoucího dopravníku a vykládány na požadované pozice na odkládací plochu. Pro potřeby úlohy byl vytvořen program v jazyce C# pro přeposílání zprávy přes sériový port.

Ke všem úlohám byly vytvořeny návody, které jsou součástí elektronické přílohy. První úloha byla testována skupinou studentů prvního ročníku navazujícího magisterského

studia oboru Robotika. Studenti neměli předchozí zkušenosti s programováním ani se softwarem RobotStudio. Při testování návodu pro programování prostřednictvím dotykového prostředí (příloha A) bylo zjištěno, že se úloha dá zvládnout za dvě hodiny. Problematická místa v návodu byla na základě zpětné vazby optimalizována. Tvorba simulace úlohy - stavba pyramidy (příloha B) – byla pro studenty intuitivnější a popsané kroky jsou v návodu dostatečně podrobné. Vytvoření simulace je zvládnutelné za čtyři hodiny.

Za hlavní částí je popsána problematika funkce Smart Component, která je obsažena v RobotStudiu. Tvorba prvků pomocí této funkce značně zrychluje tvorbu simulace a takto vytvořené komponenty se dají využít i v dalších simulacích bez ztráty vlastností (např. uchopení objektu manipulace).

## 6 Informační zdroje

[1] BYRTUS, K. *Integrace strojového vidění do robotizovaného pracoviště: diplomová práce*. Ostrava: VŠB-TUO, Univerzitní studijní programy, Studijní program Mechatronika, 2012. 59 s. Vedoucí práce: Novák, P.

[2] TOMÁŠ, J. *Realizace efektoru s přísavkami pro PR ABB IRB 140: diplomová práce*. Ostrava: VŠB-TUO, Fakulta strojní, Katedra robototechniky, 2010. 62 s. Vedoucí práce: Krys, V.

[3] SZCZEPONIEC, J. *Návrh a realizace výukové ho RTP pro PR ABB IRB 140 na UCR: diplomová práce*. Ostrava: VŠB-TUO, Fakulta strojní, Katedra robototechniky, 2010. 65 s. Vedoucí práce: Kárník, L.

[4] CHAMRAD, T. *Aplikace strojového vidění pro robotizovaná technologická pracoviště: diplomová práce*. Ostrava: VŠB-TUO, Fakulta strojní, Katedra robototechniky, 2013. 122 s. Vedoucí práce: Ing. Václav Krys, Ph.D.

[5] CHAMRAD, T. *Použití průmyslové kamery pro identifikaci barevných puků, Ostrava, VŠB – TUO, Fakulta strojní, Katedra robototechniky, 2012. 47 s.*

[6] MŽIK, J. *Postup tvorby simulace třídění puků v programu RobotStudio, Ostrava, VŠB – TUO, Fakulta strojní, Katedra robototechniky, 2012. 69 s.*

[7] SCHUNK: *GWB*. [online] s. 27 [cit.2013-10-18] Dostupné z: [http://www.schunk.com/schunk\\_files/attachments/GWB\\_gesamt\\_EN.pdf](http://www.schunk.com/schunk_files/attachments/GWB_gesamt_EN.pdf)

[8] SCHUNK: *PGN-plus*. [online] s. 13 [cit.2013-10-18] Dostupné z: [http://www.schunk.com/schunk\\_files/attachments/PGN-plus\\_80\\_En.pdf](http://www.schunk.com/schunk_files/attachments/PGN-plus_80_En.pdf)

[9] SCHUNK: *SWS*. [online]. s. 10 [cit.2013-10-18]. Dostupné z: [http://www.schunk.com/schunk\\_files/attachments/SWS\\_005\\_DE.pdf](http://www.schunk.com/schunk_files/attachments/SWS_005_DE.pdf)

[10] *ABB Group* [online]. 2013 [cit. 2013-10-11]. Dostupné z: <http://www.abb.com/>

[12] SCHUNK: *PZN-plus*. [online]. s. 10 [cit.2013-10-18]. Dostupné z: [http://www.schunk.com/schunk\\_files/attachments/PZN-plus\\_064\\_EN.pdf](http://www.schunk.com/schunk_files/attachments/PZN-plus_064_EN.pdf)

- [13] *NI Vision: NI CVS-1450 Series User Manual*. Austin, Texas, 2007. Dostupné z: <http://www.ni.com/pdf/manuals/373610f.pdf>
- [14] *NI Vision I/O Terminal Block and Prototyping Accessory*. Dostupné z: <http://www.ni.com/pdf/manuals/374051b.pdf>
- [15] *RAPID Reference Manual - System Data Types and Routines* On-line, ID: 3HAC 0966-13, Dostupné z: [http://rab.ict.pwr.wroc.pl/irb1400/datasys\\_rev1.pdf](http://rab.ict.pwr.wroc.pl/irb1400/datasys_rev1.pdf)
- [16] *RAPID Reference Manual - RAPID Overview* On-line, ID: 3HAC 0966-50 Dostupné z: <http://rab.ict.pwr.wroc.pl/irb1400/overviewrev1.pdf>
- [17] *HSL and HSV*. Wikipedia [online]. 2014 [cit. 2014-05-02]. Dostupné z: [http://en.wikipedia.org/wiki/HSI\\_color\\_space](http://en.wikipedia.org/wiki/HSI_color_space)
- [18] *HSV*. Wikipedia [online]. 2014 [cit. 2014-05-02]. Dostupné z: <http://cs.wikipedia.org/wiki/HSL>
- [19] KOZMÁK, Jiří. Barevné prostory. *Beetyho Diplomová práce* [online]. 1999 [cit. 2014-05-02]. Dostupné z: <http://komzak.webz.cz/old/skola/dp/prostory.html>
- [20] SCHUNK: *PGF*. [online]. s. 10 [cit.2013-10-18]. Dostupné z: [http://www.schunk.com/schunk\\_files/attachments/PGF\\_gesamt\\_EN.pdf](http://www.schunk.com/schunk_files/attachments/PGF_gesamt_EN.pdf)
- [21] *XUM-LH0451 Datasheet*. Datasheet Archive [online]. 2014 [cit. 2014-05-02]. Dostupné z: <http://www.datasheetarchive.com/XUM-LH0451-datasheet.html>

## 7 Seznam příloh

Všechny přílohy jsou pouze v elektronické podobě.

Příloha A:

Návod pro tvorbu programu pro vykládání kostek na dopravník pomocí dotykového rozhraní (.pdf)

Příloha B:

Stavba pyramidy

Obsahuje:

- Návod pro simulaci – stavba pyramidy (.pdf)
- Pack&Go balíček stanice v RobotStudios (.rspag)
- Záloha programů pro roboty (.prg)
- video ze simulace a reálného nasazení (.wmv)

Příloha C:

Třídění destiček s kruhovou dírou

Obsahuje:

- Návod pro simulaci – třídění destiček a návod pro CVS (.pdf)
- Pack&Go balíček stanice v RobotStudios (.rspag)
- Záloha programů pro roboty (.prg) a programu pro kamerový systém (.vbait)
- video ze simulace a reálného nasazení (.wmv)

Příloha D:

Třídění dominových kamenů

Obsahuje:

- Návod pro sestavení úlohy – třídění dominových kamenů (.pdf)
- Záloha programů pro roboty (.prg) a programu pro kamerový systém (.vbait)
- Program – Sériová komunikace (.exe) a komentovaný zdrojový kód
- video reálného nasazení (.wmv)

Příloha E:

- videonávod – Smart Component – přísavkový efektor (.avi)
- videonávod – Smart Component – čelistový efektor (.avi)